UNIVERSITÄT
DES
SAARLANDES

Saarland University
Faculty of Natural Sciences and Technology I
Department of Computer Science

A flexible framework for solving constrained ratio problems in machine learning

Thomas Bühler, M.Sc.

Dissertation

zur Erlangung des Grades
des Doktors der Naturwissenschaften (Dr. rer. nat.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Saarbrücken, Dezember 2014

Tag des Kolloquiums:
17.06.2015

Dekan:
Prof. Dr. Markus Bläser

Vorsitzender des Prüfungsausschusses:
Prof. Dr. Joachim Weickert
Universität des Saarlandes

1. Gutachter:
Prof. Dr. Matthias Hein
Universität des Saarlandes

2. Gutachter:
Prof. Dr. Ulrike von Luxburg
Universität Hamburg

Akademischer Mitarbeiter:
Dr. Moritz Gerlach
Universität des Saarlandes

Abstract

The (constrained) optimization of a ratio of non-negative set functions is a problem appearing frequently in machine learning. As these problems are typically NP hard, the usual approach is to approximate them through convex or spectral relaxations. While these relaxations can be solved globally optimal, they are often too loose and thus produce suboptimal results. In this thesis we present a flexible framework for solving such constrained fractional set programs (CFSP). The main idea is to transform the combinatorial problem into an equivalent unconstrained continuous problem. We show that such a tight relaxation exists for every CFSP. It turns out that the tight relaxations can be related to a certain type of nonlinear eigenproblem. We present a method to solve nonlinear eigenproblems and thus optimize the corresponding ratios of in general non-differentiable differences of convex functions. While the global optimality cannot be guaranteed, we can prove the convergence to a solution of the associated nonlinear eigenproblem. Moreover, in practice the loose spectral relaxations are outperformed by a large margin. Going over to constrained fractional set programs and the corresponding nonlinear eigenproblems leads to a greater modelling flexibility, as we demonstrate for several applications in data analysis, namely the optimization of balanced graph cuts, constrained local clustering, community detection via densest subgraphs and sparse principal component analysis.

Zusammenfassung

Die (beschränkte) Optimierung von nichtnegativen Bruchfunktionen über Mengen ist ein häufig auftretendes Problem im maschinellen Lernen. Da diese Probleme typischerweise NP-schwer sind, besteht der übliche Ansatz darin, sie durch konvexe oder spektrale Relaxierungen zu approximieren. Diese können global optimal gelöst werden, sind jedoch häufig zu schwach und führen deshalb zu suboptimalen Ergebnissen. In dieser Arbeit stellen wir ein flexibles Verfahren zur Lösung solcher beschränkten fraktionellen Mengenprogramme (BFMP) vor. Die Grundidee ist, das kombinatorische in ein equivalentes unbeschränktes kontinuerliches Problem umzuwandeln. Wir zeigen dass dies für jedes BFMP möglich ist. Die strenge Relaxierung kann dann mit einem nichtlinearen Eigenproblem in Bezug gebracht werden. Wir präsentieren ein Verfahren zur Lösung der nichtlinearen Eigenprobleme und damit der Optimierung der im Allgemeinen nichtdifferenzierbaren und nichtkonvexen Bruchfunktionen. Globale Optimalität kann nicht garantiert werden, jedoch die Lösung des nichtlinearen Eigenproblems. Darüberhinaus werden in der Praxis die schwachen spektralen Relaxierungen mit einem großen Vorsprung übertroffen. Der Übergang zu BFMPs und nichtlinearen Eigenproblemen führt zu einer verbesserten Flexibilität in der Modellbildung, die wir anhand von Anwendungen in Graphpartitionierung, beschränkter lokaler Clusteranalyse, dem Finden von dichten Teilgraphen, sowie dünnbesetzter Hauptkomponentenanalyse demonstrieren.

Acknowledgements

First of all, I would like to express my gratitude towards my supervisor Matthias Hein. During the development of this thesis, we had many inspiring discussions and I thank him for raising my interest in this exciting field and for his support and helpful advice. I learned a lot during my work on this thesis, for which I am very grateful.

I would also like to thank my officemate and co-author Shyam Rangapuram for many constructive discussions and his invaluable contribution to our joint projects, and last but not least for all the pleasant time we spent together in the office. I also thank Simon Setzer for inspiring conversations and our very constructive collaboration.

Special thanks go to Martin Slawski who accompanied me on this path towards the PhD from the very beginning, and who was always an inspiration in terms of discipline and perseverance. Many thanks also to Antoine Gautier for proofreading parts of this thesis. In addition, I would like to thank my former officemate Kwang In Kim for helpful discussions.

Moreover, many thanks to Alex Fauss for his invaluable work in keeping our infrastructure alive and his availability to fix urgent problems at late hours and weekends. Special thanks also go to Irmtraud Stein and Dagmar Glaser for their help in administrative tasks.

Furthermore, I would like to express my gratitude to all other current and former members of the Machine Learning Group who contributed implicitly to this thesis through inspiring discussions which influenced the way I think about Machine Learning, and by providing a great atmosphere and a pleasant place to work.

Finally I would like to thank my family for always believing in me and their constant support and encouragement during the work on this thesis.

Contents

Chapter 1

Introduction

How does one develop a machine learning method for a particular real world application? Typically, the derivation involves two steps. First, one formulates a mathematical model describing the real world scenario as accurately as possibly. Second, one designs an algorithm to solve the underlying mathematical problem efficiently. Being able to do both of those steps is crucial to the development of a useful technique for the given task.

On the one hand, it is of no use to have a model which is a perfectly accurate representation of the real world task, if there is no way to solve the underlying mathematical problem. It may be that there is no known algorithm to compute a solution efficiently, or that one can construct a numerical scheme to solve the problem approximately, but the result is far away from the true solution. Or it may be that one can prove that there is no way to obtain an optimal solution in reasonable time while at the same time being able to guarantee that it is in fact optimal.

On the other hand, it is also not helpful if there exists a mathematical description of the problem which can be solved efficiently to global optimality, but is only a coarse approximation of the real world task. This is what is commonly meant by the phrase "this does not work in practice" - making too simplifying modeling assumptions will lead to a result far away from the true solution of the practical problem.

If we have a simple model and a complex model describing a real world scenario equally well, of course the principle of Occam's razor tells us to favor the simpler solution. However, often the reason one uses a simpler formulation is not because it is sufficient to describe the practical application, but because there is no available technique for the more complex problem. For this reason, it is of crucial importance to study the underlying mathematical concepts, and in particular to advance our knowledge on how to solve certain classes of mathematical optimization problems.

The class of problems dealt with in this thesis is the optimization of ratios of two functions. There is a wide range of applications of this type of

problems. Ratios of two functions appear in situations where the goal is to optimize a trade-off between two quantities, or equivalently, minimize one quantity while maximizing another quantity simultaneously. For instance, in finance, one often encounters problems where one is interested in maximizing return while minimizing investment. Similarly, in resource allocation problems, one aims at minimizing a ratio between cost and return. Another example is given by scheduling problems where a certain ratio of cost per time is optimized. See Schaible [1981] and references therein for a number of other applications. Another example is if one is interested in finding communities in a network, for instance a social network. Here, a community can be defined as having large connectivity while having small size, hence again leading to a ratio problem [Fortunato, 2010].

Certain special classes of ratio problems have been widely studied in the literature and can be solved efficiently. These special cases are when the involved functions are linear, quadratic, or a ratio of a convex and a concave function [Dinkelbach, 1967, Schaible, 1981]. In the case of a certain class of quadratic functions, the ratio problem can be related to the solution of a linear eigenproblem. The most popular examples of this type of problems in machine learning are spectral clustering as well as principal component analysis [von Luxburg, 2007, Jolliffe, 2002].

However, the limitation to these types of functions imposes a severe modeling restriction. This forms the motivation to consider a more general class of ratio problems, where the involved non-negative functions are in general non-convex and non-differentiable. It turns out that in these cases, the problems can be related to the solution of a nonlinear eigenproblem, which will be discussed in this thesis.

Another issue arises in the presence of additional side constraints, or prior information about the optimal solution. Ideally, one would like to incorporate these constraints into the model, thus allowing for a more accurate description of the real world problem. For instance, in financial applications, one may have additional budget constraints, or in the community detection task one may have additional restrictions regarding the size of the found communities. There has been some previous work on incorporating constraints into some of the applications discussed above (see e.g. Mahoney et al. [2012], Khuller and Saha [2009], Saha et al. [2010]), however often these methods fail to guarantee that the constraints are fulfilled by the returned solution.

For this reason, in this thesis we will consider a certain type of mathematical optimization problems called *constrained fractional programs*, in particular the special case where the input argument is a set of elements, referred to as *constrained fractional set programs*. Going over to the more general class of problems will lead to an improved modeling flexibility. Moreover, we will also provide a numerical scheme to solve the resulting problem efficiently. In the following, we will discuss some applications in machine learning.

1.1 Constrained fractional (set) programs

In this section we discuss some examples of ratio problems in machine learning. We start by describing several problems in a graph-based setting. In the following, $G = (V, E, W)$ denotes an undirected, weighted graph, where V is the set of vertices and E the set of edges. Moreover, each edge is assigned a non-negative weight, where the weights are encoded in a non-negative, symmetric weight matrix $W \in \mathbb{R}^{n \times n}$, where $n = |V|$.

Graphs arise in a wide range of applications in machine learning. Often the graphs are constructed from the given data in such a way that the vertices correspond to data points and the edge weights represent pairwise similarities, see e.g. von Luxburg [2007]. In other cases, the data is already given in the form of a graph, for example in social networks, communication networks or web graphs, see e.g. Leskovec et al. [2009], Fortunato [2010].

By assigning a non-negative weight g_i to each vertex i, we can define the general volume of a subset $A \subset V$ as $\mathrm{vol}_g(A) = \sum_{i \in A} g_i$. As special cases, we obtain for $g_i = 1$ the cardinality $|A|$ and for g_i equal to the degree $d_i = \sum_{j \in V} w_{ij}$ the classical volume $\mathrm{vol}(A) = \mathrm{vol}_d(A)$. Furthermore, $\overline{A} = V \backslash A$ denotes the complement of A.

1.1.1 Balanced graph cuts

Balanced graph cuts are a well-known class of problems in computer science. The aim is to achieve a bi-partition of the graph which has only small connection between the two parts, while at the same time the partition is balanced with respect to some notion of size. This problem has applications ranging from parallel computing to image segmentation [Pothen et al., 1990, Shi and Malik, 2000]. There exist several different criteria for the balanced graph cut problem. A very popular objective is the normalized cut

$$\mathrm{NCut}(C, \overline{C}) = \frac{\mathrm{cut}(C, \overline{C})}{\mathrm{vol}_d(C)\,\mathrm{vol}_d(\overline{C})}, \quad \text{for } C \subset V,$$

where $\mathrm{cut}(C, \overline{C}) := \sum_{i \in C, j \in \overline{C}} w_{ij}$. The spectral relaxation of the normalized cut leads to the popular spectral clustering method [von Luxburg, 2007]. A related criterion with a slightly different balancing behavior is the normalized Cheeger cut,

$$\mathrm{NCC}(C, \overline{C}) = \frac{\mathrm{cut}(C, \overline{C})}{\min\{\mathrm{vol}_d(C), \mathrm{vol}_d(\overline{C})\}}, \quad \text{for } C \subset V.$$

We will discuss more balanced graph cuts in Chapter 7, and show how they can be optimized using the framework introduced in this thesis. More general balanced graph cuts were studied by Hein and Setzer [2011]. Fig. 1.1 gives an example of a balanced graph partitioning problem.

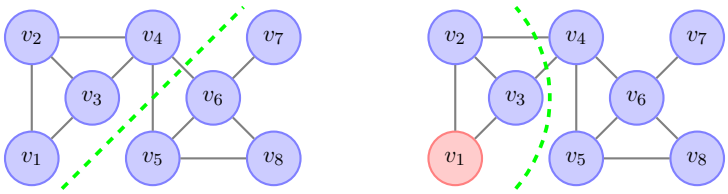

Figure 1.1: Illustration of some of the considered graph-based problems. *Left:* Balanced graph cut. *Right:* Local clustering with subset constraint.

1.1.2 Constrained balanced graph cuts for local clustering

In practice, there is often additional information available about the given task. This knowledge can be incorporated into the optimization problem in the form of constraints. In the case of graph partitioning and clustering, this leads to the problem of constrained local clustering which recently has gained some attention in the machine learning community.

Starting with the work of Spielman and Teng [2004], initially, the goal was to develop an *algorithm* that finds a subset near a given seed vertex with *small* normalized cut or normalized Cheeger cut value with running time linear in the size of the obtained cluster. The proposed algorithm and subsequent work [Andersen et al., 2006, Andersen and Peres, 2009, Oveis Gharan and Trevisan, 2012, Zhu et al., 2013] start with a given seed vertex and then use random walks to explore the graph locally, without considering the whole graph. Algorithms of this type have been applied for community detection in networks [Andersen and Lang, 2006].

In contrast, Mahoney et al. [2012] give up the runtime requirement and formulate the task as an explicit optimization problem, where the goal is to find the *optimal* normalized cut subject to a seed constraint and an upper bound on the volume of the returned set. They then derive a spectral-type relaxation of the normalized cut problem which is biased towards solutions fulfilling the seed constraint. Their method has been applied in semi-supervised image segmentation [Maji et al., 2011] and for community detection around a given query set [Mahoney et al., 2012]. However, while they provide an approximation guarantee for their relaxation, they cannot guarantee that the returned solution satisfies seed and volume constraints.

In Chapter 8 we consider an extended version of the problem of Mahoney et al. [2012]. Let J denote the set of seed vertices, and $\widehat{S}$ a symmetric balancing function (e.g. $\widehat{S}(C) = \operatorname{vol}_d(C)\operatorname{vol}_d(\overline{C})$ for the normalized cut). The general local clustering problem can then be formulated as

$$\begin{aligned} &\min_{C \subset V} \frac{\operatorname{cut}(C, \overline{C})}{\widehat{S}(C)} \\ &\text{subject to}: \ \operatorname{vol}_h(C) \leq k, \quad \text{and} \quad J \subset C, \end{aligned} \tag{1.1}$$

where $h \in \mathbb{R}^n_+$ are vertex weights. An example can be found in Fig. 1.1, where we specified a seed vertex as well as an upper bound constraint of the form $|C| \leq 3$. The choice of the balancing function $\widehat{S}$ allows the user to influence the trade-off between getting a partition with small cut and one with good balance.

1.1.3 Constrained local community detection

A related problem is constrained local community detection. In community detection it makes more sense to find a highly connected set instead of emphasizing the separation to the remaining part of the graph. Thus, we are searching for a set C which has high association, defined as $\text{assoc}(C) = \sum_{i,j \in C} w_{ij}$. Dividing the association of C by its size yields the density of C. The subgraph of maximum density can be computed in polynomial time [Goldberg, 1984]. However, the obtained communities in the unconstrained problem are often too large or too small, thus one is required to incorporate additional size constraints. Unfortunately, the introduction of such constraints makes the problem NP-hard [Khuller and Saha, 2009].

In Chapter 9 we will consider a general class of (local) community detection problems, which can be formulated as

$$\begin{aligned} &\max_{C \subset V} \frac{\text{assoc}(C)}{\text{vol}_g(C)} \\ &\text{subject to} : k_1 \leq \text{vol}_h(C) \leq k_2, \quad \text{and} \quad J \subset C, \end{aligned} \tag{1.2}$$

where $g, h \in \mathbb{R}^n_+$ are vertex weights. This formulation generalizes the above-mentioned density-based approaches by replacing the denominator by a general volume function vol_g. The use of vertex weights allows us to bias the obtained community towards one with desired properties (assigning small weights to vertices which one prefers to be contained in the solution, larger weights to ones which are less preferred).

The special case of (1.2) where one has only lower bound constraints has been considered in team selection [Gajewar and Das Sarma, 2012] and bioinformatics [Saha et al., 2010], where constant factor approximation algorithms have been derived. However, in the case of equality and upper bound constraints, the problem is very hard. It has been shown that even when using only cardinality constraints (i.e. $h_i = 1$), there exists no polynomial time approximation scheme [Khot, 2006, Khuller and Saha, 2009].

Our method can handle such hard upper bound and equality constraints, as we will show in Chapter 9. While no approximation guarantees can be given, excellent qualitative results are achieved in practice, which will be demonstrated in the experiments for a community detection problem with a specified query set J and an upper bound on the size for a co-author network as well as a composer network.

1.1.4 Sparse principal component analysis (PCA)

We now leave the graph-based setting and consider an example of a ratio problem where the optimization is done over $\mathbb{R}^n$. Principal component analysis (PCA) is a standard technique for dimensionality reduction and data analysis [Jolliffe, 2002]. PCA finds the p-dimensional subspace of maximal variance in the data. For $p = 1$, this can be formulated as

$$\max_{f \in \mathbb{R}^n} \frac{\langle f, \Sigma f \rangle}{\|f\|_2^2} ,$$

where $\Sigma \in \mathbb{R}^{n \times n}$ is the sample covariance matrix of the given data. As usually all entries of the optimum of the above problem are nonzero, an interpretation of the principal components is often difficult. This plays a role for instance in the case of gene expression data where one would like the loading vectors of the principal components to consist only of a few significant genes, making it easy to interpret by a human.

For this reason, in sparse PCA one enforces sparsity of the solution with the aim of getting a small number of features while at the same time still capturing most of the variance. In other words, one is interested in the optimal trade-off between explained variance and sparsity. In Section 10 we show how the sparse PCA problem can be modeled as constrained fractional program and solved using the techniques introduced in this thesis.

1.2 Loose convex vs. tight non-convex relaxations

Note that the combinatorial problems considered in this thesis are in general NP-hard [Šíma and Schaeffer, 2006, Shi and Malik, 2000, Feige et al., 2001, Khuller and Saha, 2009, Moghaddam et al., 2006]. Thus the standard approach is to replace them by convex or spectral relaxations which can be solved globally optimally. The spectral relaxation is very popular in machine learning, e.g. spectral clustering [Hagen and Kahng, 1991, Shi and Malik, 2000]. However, it is often quite loose and thus leads to a solution far away from the optimal one of the original problem. Moreover, spectral-type relaxations [Mahoney et al., 2012] fail to guarantee that the constraints which encode the prior knowledge are satisfied.

A different approach is to consider tight non-convex relaxations instead. Here, tight relaxation means that the continuous and the combinatorial optimization problem are equivalent in the sense that the optimal values agree and the optimal solution of the combinatorial problem can be obtained from the continuous solution (and vice versa). In a recent line of work [Hein and Bühler, 2010, Szlam and Bresson, 2010, Hein and Setzer, 2011, Bresson et al., 2012a], it has been shown that tight continuous relaxations exist for all balanced graph cut problems and the normalized cut subject to must-link and cannot-link constraints [Rangapuram and Hein, 2012].

The obvious disadvantage of these types of approaches is that they provide no guarantee to yield the globally optimal solution. However, in practice the standard loose relaxations are outperformed by a large margin. Moreover, in contrast to the loose relaxations, tight relaxations guarantee that all constraints are satisfied. In this thesis we show that tight relaxations exist for all constrained fractional set programs. In particular this allows us to derive efficient methods for the ratio problems discussed in the last section.

1.3 Overview of this thesis

The thesis is structured into three parts: Part I deals with the theoretical foundations of constrained fractional set programs and their relation to nonlinear eigenproblems. We start in Chapter 2 by reviewing basic concepts such as set functions, submodularity and Lovász extensions. Chapter 3 defines nonlinear eigenproblems and shows their connection to critical points of ratios of non-negative functions. Chapter 4 shows that any fractional set program (constrained or unconstrained) can be transformed into an equivalent optimization problem involving a ratio of non-negative functions.

Part II deals with the optimization of such ratios of non-negative functions. In Chapter 5 we discuss several different cases, depending on the class of functions R and S. Note that the ratio is in general non-convex and non-differentiable. The most general case considered is the case where R and S are non-negative differences of convex functions, thus covering a wide range of different problems. The main idea of all methods discussed in this chapter is to decompose the non-convex problem into a sequence of convex problems which can be solved globally optimally. In Chapter 6 we discuss how to solve this convex inner problem efficiently. We give an overview about various methods for convex optimization and then propose a general-purpose method to solve the inner problem based on bundle methods.

Finally, in Part III we use the theoretical results to develop several applications in data analysis. Chapter 7 deals with balanced graph cuts in particular the optimization of the Cheeger cut objective, which is used in clustering. We discuss the methods p-spectral clustering and 1-spectral clustering and show their superiority to state of the art methods. Chapter 8 presents a method for the problem of constrained local clustering. We apply the method on several large social network datasets. Chapter 9 considers the related problem of community detection based on the constrained maximum density subgraph problem. We develop a method based on tight relaxations of constrained fractional set programs and demonstrate its ability to detect meaningful communities in a network of computer science researchers as well as a network of classical composers. Chapter 10 develops a method for sparse PCA, which matches state of the art result on gene expression data sets as well as a well-known sparse PCA benchmark dataset.

1.3.1 Main contributions

In Chapter 4 we show that all constrained non-negative fractional set programs have an equivalent tight continuous relaxation. Note that the results in [Hein and Setzer, 2011, Rangapuram and Hein, 2012] are not applicable to all considered problems because of two limitations: First, tight relaxations were shown only for a ratio of symmetric non-negative set functions, where the numerator is restricted to be submodular. Second, only equality constraints for non-negative submodular or supermodular set functions could be handled. We extend the results to arbitrary ratios of non-negative set functions with inequality constraints, without any further restrictions.

In Chapter 5 we present a nonlinear inverse power method to compute the solutions of nonlinear eigenproblems, which is a necessary condition for critical points of the associated nonlinear Rayleigh quotient. The nonlinear IPM allows us to handle several ratio problems discussed in this thesis. After the initial publication of this result [Hein and Bühler, 2010], the method was further generalized to arbitrary ratios of non-negative 1-homogeneous differences of convex functions by Hein and Setzer [2011]. The more general method allows us to tackle all remaining problems in this thesis.

In Chapter 6 we give a general-purpose method for the convex inner problem which does not require a closed form of the involved Lovász extensions and uses only evaluations of the original set function in each step.

In Chapter 7 we propose 1-spectral clustering, a method based on nonlinear eigenproblems for the problem of finding an optimal Cheeger cut on a graph. Moreover, we consider a variation based on the symmetric vertex expansion of the graph. We show that our methods converge to an eigenvector of the associated nonlinear eigenproblem and consistently outperform competing methods [von Luxburg, 2007, Szlam and Bresson, 2010] in terms of the obtained objective value and quality of the obtained clustering.

In Chapter 8 we present an efficient method for the problem of local clustering with volume and seed constraints. Our algorithm consistently outperforms competing methods [Andersen and Lang, 2006, Mahoney et al., 2012]. Moreover, we are not aware of any other methods for this problem which can guarantee that the solution always satisfies volume and seed constraints.

In Chapter 9 we present a method for the problem of community detection in a network by finding a maximum density subgraph in the graph subject to size constraints. We demonstrate the usefulness of the method by its ability to detect meaningful communities in a network of computer science researchers as well as a network of classical composers.

In Chapter 10 we present a method for sparse principal component analysis, which matches the performance of state of the art methods [Zou et al., 2006, Sigg and Buhmann, 2008, Journée et al., 2010] as demonstrated on several gene expression data sets as well as a well-known sparse PCA benchmark dataset.

Part I

Theoretical foundations of constrained fractional set programs and nonlinear eigenproblems

Chapter 2

(Submodular) set functions and Lovasz extensions

In this chapter we establish the mathematical groundwork for the results in the remainder of the thesis. We start by reviewing some basic concepts from analysis such as convex functions, the subdifferential of a convex function as well as p-homogeneity. Next we consider set functions and their continuous extensions, in particular the so-called Lovász extension which will play a major role in this thesis. Finally, we cover a special class of set functions called submodular functions, which are important due to their connection to convex functions. The concepts introduced in this chapter will later be used to show that every constrained fractional set program has a tight relaxation into a continuous fractional program.

2.1 Basics from analysis

We begin by reviewing some basic definitions from convex analysis, see for example Rockafellar [1970]. Two essential mathematical concepts are convex sets and convex functions.

Definition 2.1 (Convex set). *A set $C \subset \mathbb{R}^n$ is a convex set if for all $f, g \in C$ and all $\alpha \in [0,1]$ it holds that $\alpha f + (1-\alpha)g \in C$.*

Definition 2.2 (Convex function). *A function $R : \mathbb{R}^n \to \mathbb{R}$ is a convex function if for all $f, g \in \mathbb{R}^n$ and all $\alpha \in [0,1]$ it holds that*

$$R(\alpha f + (1-\alpha)g) \;\leq\; \alpha R(f) + (1-\alpha)R(g). \tag{2.1}$$

Convex functions have the useful property that every local minimum is a global minimum. Thus there exists a wide range of methods for the globally optimal solution of *convex minimization problems* (i.e. the minimization of a convex function over a convex set), see e.g. Boyd and Vandenberghe [2004]. In the following we give two special cases of convexity.

Definition 2.3 (Strict and strong convexity). *A function $R : \mathbb{R}^n \to \mathbb{R}$ is strictly convex if the inequality in (2.1) is strict for all $f, g \in \mathbb{R}^n$ with $f \neq g$ and all $\alpha \in (0, 1)$. R is strongly convex if there exists a parameter $\mu > 0$ such that for all $f, g \in \mathbb{R}^n$ and all $\alpha \in [0, 1]$ it holds that*

$$R(\alpha f + (1-\alpha)g) \;\leq\; \alpha R(f) + (1-\alpha)R(g) - \frac{\mu}{2}\alpha(1-\alpha)\left\|f-g\right\|_2^2 .$$

Clearly, every strongly convex function is also strictly convex. Moreover, one easily checks that the above definition of strong convexity is equivalent to saying that the function $R(f) - \frac{\mu}{2}\left\|f\right\|_2^2$ is convex. The functions considered in this thesis will in general be non-differentiable. An important tool to deal with such functions is the subdifferential of a convex function, which is a generalization of the gradient to the non-differentiable case.

Definition 2.4 (Subdifferential). *Let $R : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Then the subdifferential at a point f is defined as*

$$\partial R(f) = \{r \in \mathbb{R}^n : R(g) \geq R(f) + \langle r, g - f\rangle , \forall g \in \mathbb{R}^n\}.$$

An element $r \in \partial R(f)$ is called subgradient.

Geometrically, this means that every subgradient r at a point f defines a hyperplane with normal vector $(r, -1)$ which supports the epigraph of R at the point $(f, R(f))$. A related concept is the convex conjugate of R.

Definition 2.5 (Convex conjugate). *For a function $R : \mathbb{R}^n \to \mathbb{R}$, the convex conjugate $R^* : \mathbb{R}^n \to \mathbb{R}$ is defined as*

$$R^*(y) = \sup\{\langle y, x\rangle - R(x) \mid x \in \mathbb{R}^n\}.$$

If R is differentiable, then $\partial R(f)$ has one unique element, the gradient $\nabla R(f)$. At the globally optimal point of a convex differentiable function R one has $\nabla R(f) = 0$. Similarly, for a general non-differentiable convex function R, at the globally optimal point f it holds that $0 \in \partial R(f)$. The following is a generalization of convex functions.

Definition 2.6 (Quasi-convex function). *A function $R : \mathbb{R}^n \to \mathbb{R}$ is quasi-convex if its sublevel sets $S_\alpha = \{f \in \mathbb{R}^n \mid R(f) \leq \alpha\}$ are convex.*

In contrast to convex functions, quasi-convex functions may have local minima which are not globally optimal. However, minimization problems involving quasi-convex functions can still be solved globally optimally using the convexity of their sublevel sets, which allows us to transform them into a sequence of convex feasibility problems [Boyd and Vandenberghe, 2004].

Another class of functions which will frequently be encountered in this thesis are positively p-homogeneous functions, which are functions with multiplicative scaling behavior.

Definition 2.7 (*p*-homogeneity). *A function $R : \mathbb{R}^n \to \mathbb{R}$ is positively p-homogeneous for $p > 0$ if $R(\alpha f) = \alpha^p R(f)$, $\forall \alpha \in \mathbb{R}$ with $\alpha \geq 0$.*

The relation between p-homogeneous functions and their subdifferential is described in the generalized Euler identity, see Yang and Wei [2008].

Lemma 2.8. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be a convex continuous and positively p-homogeneous function. Then, for each $f \in \mathbb{R}^n$ and $r \in \partial R(f)$ and each $p > 0$ it holds that $\langle f, r \rangle = p\, R(f)$.*

In the case of 1-homogeneous convex functions, one can give the following characterization which will be useful later.

Lemma 2.9. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be a convex continuous and positively 1-homogeneous function. Then one has $\forall f \in \mathbb{R}^n$, $R(f) = \sup_{u \in U} \langle u, f \rangle$, where the convex set U is given as $U := \partial R(0) = \{u \in \mathbb{R}^n \,|\, R(g) \geq \langle u, g \rangle \ \forall g \in \mathbb{R}^n\}$. Moreover, $\forall f \in \mathbb{R}^n$, $\partial R(f) \subset \partial R(0)$.*

Proof. Note that due to the 1-homogeneity of R, we must have $R(0) = 0$, and thus one obtains using Def. 2.4,

$$\partial R(0) = \{u \in \mathbb{R}^n \mid \forall g \in \mathbb{R}^n : \ R(g) \geq R(0) + \langle u, g - 0 \rangle = \langle u, g \rangle\}\,.$$

The convexity of U can be obtained using Def. 2.1. Moreover, $\forall f \in \mathbb{R}^n$,

$$u \in \partial R(f) \Rightarrow \forall g \in \mathbb{R}^n : \ R(g) \geq R(f) + \langle u, g - f \rangle = \langle u, g \rangle\,,$$

where we have used Lemma 2.8. Thus, $u \in \partial R(0)$, and therefore $\partial R(f) \subset \partial R(0)$. Finally, due to Lemma 2.8 one has $R(f) = \langle r, f \rangle$ for all $r \in \partial R(f) \subset U$, which implies that $R(f) = \sup_{u \in U} \langle u, f \rangle$. □

2.2 Set functions and their extensions

In the following, let V denote an arbitrary finite ground set of elements, for instance a set of features, a set of vertices of a graph, or a set of pixels of an image.

Definition 2.10 (Set function). *Given a finite ground set of elements V, a set function is a function defined on the power set of V, i.e. $\widehat{R} : 2^V \to \mathbb{R}$.*

Set functions arise naturally in a wide range of applications. Depending on the type of ground set V, a set function could be for example an optimality criterion for a feature selection problem, a function measuring the density of a set of vertices in a graph, or a function assigning a score to a set of pixels of an image, for instance in an object detection task.

Assume that the elements of the ground set are enumerated in a certain way, i.e. $V = \{v_1, \dots v_n\}$, where $|V| = n$. Then every set $C \subset V$ can be represented by a vector $\mathbf{1}_C \in \{0,1\}^n$ which is 1 at entry j if $v_j \in C$ and 0 otherwise. We refer to the vector $\mathbf{1}_C$ as the indicator vector of the set C. By identifying a set with its indicator vector, a set function can be seen as a function defined on the corners of the hypercube, i.e. $\{0,1\}^n$, which motivates the term *pseudo-boolean function* often found in the literature, see e.g. Boros and Hammer [2002]. One can now extend the function to the continuous space $\mathbb{R}^n$ by finding a function which agrees with the original set function on the indicator vectors.

Definition 2.11 (Extension of a set function). *Let $\widehat{R} : 2^V \to \mathbb{R}$ be a set function. A function $R : \mathbb{R}^n \to \mathbb{R}$ is called an extension of $\widehat{R}$ if $\forall A \in 2^V$ it holds that $R(\mathbf{1}_A) = \widehat{R}(A)$.*

A key tool for the derivation of the results of this thesis is the *Lovász extension*, a certain way of extending a set function to the continuous space which has some useful properties, as we will discuss below. The connection between sets and elements of the continuous space is achieved via *thresholding*. Let $f \in \mathbb{R}^n$, and we assume wlog that f is ordered in ascending order $f_1 \leq f_2 \leq \dots \leq f_n$. One defines the sets

$$C_i := \{j \in V | f_j \geq f_i\}, \qquad i = 1, \dots, n. \tag{2.2}$$

Using the above sets, the Lovász extension can be defined as follows, see e.g. Lovász [1983],Bach [2013].

Definition 2.12 (Lovász extension). *Let $\widehat{R} : 2^V \to \mathbb{R}$ be a set function with $\widehat{R}(\emptyset) = 0$, and $f \in \mathbb{R}^n$ be ordered in ascending order, $f_1 \leq f_2 \leq \dots \leq f_n$. The Lovász extension $R : \mathbb{R}^n \to \mathbb{R}$ of $\widehat{R}$ is defined as*

$$\begin{aligned} R(f) &= \sum_{i=1}^{n-1} \widehat{R}(C_{i+1}) \left(f_{i+1} - f_i\right) + \widehat{R}(V) f_1 \\ &= \sum_{i=1}^{n-1} \left(\widehat{R}(C_i) - \widehat{R}(C_{i+1})\right) f_i + \widehat{R}(C_n) f_n. \end{aligned}$$

Note that $R(\mathbf{1}_C) = \widehat{R}(C)$ for all $C \subset V$, i.e. R is indeed an extension of $\widehat{R}$ from 2^V to $\mathbb{R}^n$. The equivalence between the two definitions can be shown by a reordering of terms. Throughout this thesis, we always use the hat-symbol ($\widehat{\ }$) to denote set functions and omit it for an extension to the continuous space. Moreover, in some cases the given context will require us to distinguish between Lovász extensions and other extensions. In this situation we will use the superscript ^{L} to mark the Lovász extension, e.g. use R^L for the Lovász extension of $\widehat{R}$ and R for a different non-Lovász extension. However, we will omit the superscript ^{L} if no confusion is possible.

2.2.1 Properties of the Lovász extension

We now give some properties of Lovász extensions which will be used in the remainder of this thesis, see for example Fujishige [2005], Bach [2013]. The next proposition follows directly from the definition of the Lovász extension.

Proposition 2.13. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be the Lovász extension of $\widehat{R} : 2^V \to \mathbb{R}$. Then R is positively 1-homogeneous.*

The following proposition will be used in Chapter 4 to guarantee non-negativity of the Lovász extension. Again the proof follows in a straightforward way from the definition.

Proposition 2.14. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be the Lovász extension of $\widehat{R} : 2^V \to \mathbb{R}$. Then it holds that*

$$R(f) \geq 0, \forall f \in \mathbb{R}^n_+ \quad \textbf{\textit{iff}} \quad \widehat{R}(A) \geq 0,\ \forall A \subset V,$$
$$R(f) \geq 0, \forall f \in \mathbb{R}^n \text{ and } R(\mathbf{1}) = 0 \quad \textbf{\textit{iff}} \quad \widehat{R}(A) \geq 0, \forall A \subset V \text{ and } \widehat{R}(V) = 0.$$

The following proposition gives a way to compute Lovász extensions of set functions which can be decomposed into more elementary set functions.

Proposition 2.15. *Let $R, S : \mathbb{R}^n \to \mathbb{R}$ be the Lovász extensions of $\widehat{R}, \widehat{S} : 2^V \to \mathbb{R}$. Then, $\lambda_1 R + \lambda_2 S$ is the Lovász extension of $\lambda_1 \widehat{R} + \lambda_2 \widehat{S}$, for all $\lambda_1, \lambda_2 \in \mathbb{R}$.*

The importance of the Lovász extension arises due to its connection to submodular functions, which will be discussed in the next section.

2.3 Submodular set functions

Recently there has been a strong interest in methods based on *submodular set functions* in machine learning and related areas, with applications for example in dictionary selection [Krause and Cevher, 2010], sensor placement [Krause et al., 2008], learning graphical models [Narasimhan and Bilmes, 2004] and computer vision [Boykov et al., 2001]. The popularity of submodular functions is mainly due to their connection to convex functions, which enables them to be minimized exactly, as we will see below. Let us first discuss several equivalent definitions of submodularity, see Bach [2013].

Definition 2.16 (Submodularity). *A set function $\widehat{R} : 2^V \to \mathbb{R}$ is submodular if for all $A, B \subset V$,*

$$\widehat{R}(A \cup B) + \widehat{R}(A \cap B) \leq \widehat{R}(A) + \widehat{R}(B). \tag{2.3}$$

If the converse inequality holds, the function $\widehat{R}$ is called supermodular. It is called modular if we have equality in (2.3).

The definition also implies that $\widehat{R}$ is submodular if and only if $-\widehat{R}$ is supermodular. We give an alternative definition of submodularity (supermodularity analogously). A proof of the equivalence can be found in Bach [2013].

Proposition 2.17 (Def. with first order differences). *The set function $\widehat{R}$ is submodular if and only if for all $A, B \subset V$ and $k \in V$, such that $A \subset B$ and $k \notin B$, we have $\widehat{R}(A \cup \{k\}) - \widehat{R}(A) \leq \widehat{R}(B \cup \{k\}) - \widehat{R}(B)$.*

The above property states that submodular functions have the "diminishing returns" property, which means that the change when adding an element to a set decreases from a set A to B if $A \subset B$. In this respect, submodular functions behave like concave functions [Bach, 2013]. A third way to define submodular functions is given as follows.

Proposition 2.18 (Def. with second order differences). *The set function $\widehat{R}$ is submodular if and only if for all $A \subset V$ and $j, k \in V \backslash A$, we have $\widehat{R}(A \cup \{k\}) - \widehat{R}(A) \leq \widehat{R}(A \cup \{j, k\}) - \widehat{R}(A \cup \{j\})$.*

In practice one either shows submodularity of a given set function by using any of these definitions, or one uses the connection to convex functions which is given below (see for example Bach, 2013).

Proposition 2.19. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be the Lovász extension of $\widehat{R} : 2^V \to \mathbb{R}$. Then, $\widehat{R}$ is submodular if and only if R is convex. Furthermore, if $\widehat{R}$ is submodular, then $\min_{A \subset V} \widehat{R}(A) = \min_{f \in [0,1]^n} R(f)$.*

The above proposition implies that a submodular minimization problem can be transformed into an equivalent convex minimization problem, and thus be solved exactly. A similar equivalence between continuous and combinatorial optimization problems is shown in Chapter 4 for general ratios of non-negative set functions.

In the derivations later in this thesis, we will frequently encounter the following situation: Given a submodular set function $\widehat{R}$ and its Lovász extension R, one needs to compute an element of the subdifferential of R. However, often the Lovász extension is not available in a closed form, which makes the computation of its subdifferential difficult. A remedy for this problem is given by the following lemma, which allows us to express the subdifferential of R directly in terms of the original set function $\widehat{R}$. Moreover, one can also express R directly in terms of its subdifferential.

Lemma 2.20. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be the Lovász extension of the submodular set function $\widehat{R} : 2^V \to \mathbb{R}$. Then, an element $r(f) \in \partial R(f)$ of the subdifferential of R is given as*

$$r(f)_i = R(C_i) - R(C_{i+1}), \quad \forall i = 1, \ldots, n, \tag{2.4}$$

where the sets C_i are given in (2.2). Moreover, one has $R(f) = \langle f, r(f) \rangle$.

Proof. By Prop. 3.2 in [Bach, 2013], for all $f \in \mathbb{R}^n$ the Lovász extension R can be written as $R(f) = \max_{s \in B(\widehat{R})} \langle s, f \rangle$, where $B(\widehat{R})$ is the associated base polyhedron $B(\widehat{R})$, defined as

$$B(\widehat{R}) = \{s \in \mathbb{R}^n \mid \forall A \subset V : \langle s, \mathbf{1}_A \rangle \leq \widehat{R}(A) \text{ and } \langle s, \mathbf{1} \rangle = \widehat{R}(V)\},$$

and a maximizer is given by $r(f)$ as in (2.4). Therefore, for a given $f \in \mathbb{R}^n$ it holds for all $f' \in \mathbb{R}^n$,

$$R(f) + \langle r(f), f' \rangle - \langle r(f), f \rangle = \langle r(f), f' \rangle \leq \max_{s \in B(\widehat{R})} \langle s, f' \rangle = R(f'),$$

and thus $r(f) \in \partial R(f)$. □

The following result has been shown in Hein and Setzer [2011].

Proposition 2.21. *Every set function $\widehat{S}$ with $\widehat{S}(\emptyset) = 0$ can be written as $\widehat{S} = \widehat{S}_1 - \widehat{S}_2$, where S_1 and S_2 are submodular and $\widehat{S}_1(\emptyset) = \widehat{S}_2(\emptyset) = 0$. The Lovász extension S can be written as difference of convex functions.*

Note that while the condition $\widehat{S}_1(\emptyset) = \widehat{S}_2(\emptyset) = 0$, which is necessary for the Lovász extensions of $\widehat{S}_1$ and $\widehat{S}_2$ to be defined, is not explicitly stated in Hein and Setzer [2011], their proof ensures that it can always be fulfilled. Moreover, note that the proof is constructive, thus the decomposition into a difference of two submodular functions can always be computed. The last statement then follows using Prop. 2.15 and Prop. 2.19.

2.3.1 Examples of submodular set functions

We conclude this section by discussing some examples of submodular set function and their Lovász extensions which will be featured prominently in this thesis. There is a large number of other examples, for example flows, set covers and entropies, see for example Bach [2013] for an overview.

Generalized volume functions. The first example are functions of the form $\mathrm{vol}_g(A) = \sum_{i \in A} g_i$, where $g_i \in \mathbb{R}_+$. Generalized volume functions will play a major role in the graph-based setting considered later in this thesis. Here, the functions correspond to functions assigning a non-negative weight to each vertex. Examples include the cardinality, $|A| = \sum_{i \in A} 1$, and the classical volume, $\mathrm{vol}(A) = \sum_{i \in A} d_i$, where d_i denotes the degree of each vertex, i.e. $d_i = \sum_{j \in V} w_{ij}$ for the given graph.

Proposition 2.22. *The generalized volume function $\mathrm{vol}_g(A)$ is modular. Moreover, its Lovász extension is given by $\langle g, f \rangle$.*

Proof. Using the second formulation of the Lovász extension in Def. 2.12, one obtains the Lovász extension $R(f) = \sum_{i=1}^{n-1} (\mathrm{vol}_g(C_i) - \mathrm{vol}_g(C_{i+1}))\, f_i +$

$\mathrm{vol}_g(C_n) f_n = \sum_{i=1}^{n-1} g_i \ f_i + g_n f_n = \langle g, f \rangle$. The modularity of vol_g follows with Prop. 2.19 from the fact that $\langle g, f \rangle$ is linear. □

Cut function. Again in a graph-based setting, the cut function measures the sum of edge weights between two sets of vertices and is given as

$$\mathrm{cut}(C, \overline{C}) = \sum_{i \in C, j \in \overline{C}} w_{ij}.$$

Proposition 2.23. *The* cut *function is submodular. Moreover, its Lovász extension is given as* $\frac{1}{2} \sum_{ij \in V} w_{ij} \, |f_i - f_j|$.

Proof. The first formulation of the Lovász extension in Def. 2.12 yields

$$\begin{aligned} R(f) &= \sum_{i=1}^{n-1} \widehat{R}(C_{i+1}) \, (f_{i+1} - f_i) + \widehat{R}(C_1) f_1 \\ &= \sum_{i=1}^{n-1} \left(\sum_{k,l=1}^{n} w_{k,l} \ \delta_{k \geq i+1} \ \delta_{l \leq i} \right) \ (f_{i+1} - f_i) + \left(\sum_{k,l=1}^{n} w_{k,l} \ \delta_{k \geq 1} \ \delta_{l \leq 0} \right) f_1, \end{aligned}$$

where we used the notation $\delta_A = 1$ if A holds, and 0 else. By exchanging sums in the first term and using the fact that the last term is zero, one can rewrite this as

$$\begin{aligned} R(f) &= \sum_{k,l=1}^{n} w_{k,l} \ \sum_{i=1}^{n-1} \delta_{k \geq i+1} \ \delta_{l \leq i} \ (f_{i+1} - f_i) = \sum_{k,l=1}^{n} w_{k,l} \ \delta_{k > l} \ \sum_{i=l}^{k-1} (f_{i+1} - f_i) \\ &= \sum_{k,l=1}^{n} w_{k,l} \ \delta_{k > l} \ (f_k - f_l) = \tfrac{1}{2} \sum_{k,l=1}^{n} w_{k,l} \ |f_k - f_l| \,, \end{aligned}$$

where in the last step we have used the symmetry of W. The submodularity of $\mathrm{cut}(C, \overline{C})$ follows with Prop. (2.19) from the fact that $\frac{1}{2} \sum_{ij \in V} w_{ij} \, |f_i - f_j|$ is convex. □

The above Lovász extension is often referred to as total variation in the literature. As it favors piecewise constant solutions, it is a popular regularizer in signal processing [Rudin et al., 1992, Chambolle, 2004].

Chapter 3

Nonlinear eigenproblems

Standard eigenproblems are a well studied class of problems in linear algebra, see e.g. Horn and Johnson [1990]. While the solutions of standard eigenproblems can be related to the critical points of certain ratios of quadratic functions, it turns out that the critical points of general ratios of non-negative functions can be related to the solution of so-called *nonlinear eigenproblems*. In this chapter we will introduce nonlinear eigenproblems and discuss their relation to critical points of ratios of non-negative functions. We start by reviewing some well-known results about standard eigenproblems.

3.1 Standard (linear) eigenproblems

Given a linear mapping between two vector spaces, an eigenvector is a non-zero vector which does not change its direction under the mapping but instead yields a rescaled solution of the original vector. The scaling factor is called eigenvalue. Formally, one can give the following definition.

Definition 3.1 (Linear eigenproblem). *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, then the standard eigenproblem is the problem of finding a non-zero vector $u \in R^n$ and a scalar $\lambda \in \mathbb{R}$ satisfying the equation*

$$Au = \lambda u. \tag{3.1}$$

The vector u is called eigenvector of A with corresponding eigenvalue λ.

It is a well-known result from linear algebra that the eigenvectors of a symmetric matrix A can be characterized as critical points of the functional

$$F(f) = \frac{\langle f, Af\rangle}{\|f\|_2^2}, \tag{3.2}$$

the so-called *Rayleigh quotient*. The following theorem formalizes this connection, see e.g. Horn and Johnson [1990].

Theorem 3.2 (Rayleigh-Ritz). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix, then the smallest and largest eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$ of A are given as*

$$\lambda_{\min} = \lambda_1 = \min_{u\in\mathbb{R}^n, u\neq 0} \frac{\langle u, Au\rangle}{\|u\|_2} = \min_{u\in\mathbb{R}^n, \|u\|_2=1} \langle u, Au\rangle ,$$
$$\lambda_{\max} = \lambda_n = \max_{u\in\mathbb{R}^n, u\neq 0} \frac{\langle u, Au\rangle}{\|u\|_2} = \max_{u\in\mathbb{R}^n, \|u\|_2=1} \langle u, Au\rangle ,$$

and the minimizing/maximizing arguments $u_{\min}$ and $u_{\max}$ are the corresponding eigenvectors. Moreover, given eigenvalues $\lambda_1 \leq \cdots \leq \lambda_{k-1}$ and corresponding eigenvectors $u_1, \ldots, u_{k-1}$, the k-th eigenvalue is given as

$$\lambda_k = \min_{\substack{u\in\mathbb{R}^n, u\neq 0 \\ u\perp u_1,\ldots u_{k-1}}} \frac{\langle u, Au\rangle}{\|u\|_2} = \min_{\substack{u\in\mathbb{R}^n, \|u\|_2=1 \\ u\perp u_1,\ldots u_{k-1}}} \langle u, Au\rangle .$$

The above theorem can be used for an explicit construction of a sequence of eigenvectors. In the following, denote by $\mathcal{U}_k$ the set of all k-dimensional subspaces of $\mathbb{R}^n$. A different characterization of the k-th eigenvalue is the following, see e.g. Horn and Johnson [1990].

Theorem 3.3 (Courant-Fischer). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$, and $k \in \{1, \ldots, n\}$. Then the k-th eigenvalue of the matrix A is given as*

$$\lambda_k = \max_{U_{k-1}\in\mathcal{U}_{k-1}} \min_{\substack{u\in\mathbb{R}^n, u\neq 0 \\ u\perp U_{k-1}}} \frac{\langle u, Au\rangle}{\|u\|_2}.$$

The difference between the two formulations is that in Theorem 3.3, the characterization of the k-th eigenvalue λ_k does not require an explicit knowledge of the eigenvectors corresponding to the other eigenvalues. However, from an algorithmic point of view, the formulation is not useful, since it is intractable to optimize over all possible subspaces. We will come back to this issue in the next section when discussing the generalization to nonlinear eigenproblems. Another characterization of the k-th eigenvalue is the following, see Drábek and Milota [2007].

Theorem 3.4 (Courant-Weinstein). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$, and $k \in \{1, \ldots, n\}$. Then the k-th eigenvalue of the matrix A is given as*

$$\lambda_k = \min_{U_k\in\mathcal{U}_k} \max_{u\in U_k} \frac{\langle u, Au\rangle}{\|u\|_2}.$$

While the ratio of quadratic functions is useful in several applications, it is a severe modeling restriction. Thus in the following section we will go over to a more general class of ratios.

3.2 Nonlinear eigenproblems

In the following we consider ratios of the form

$$Q(f) \;=\; \frac{R(f)}{S(f)} \;=\; \frac{R_1(f) - R_2(f)}{S_1(f) - S_2(f)}, \tag{3.3}$$

where $R = R_1 - R_2$ and $S = S_2 - S_2$ are assumed to be non-negative differences of convex functions which are Lipschitz continuous, even (i.e. $R(f) = R(-f)$ for all $f \in \mathbb{R}^n$) and positively p-homogeneous for $p > 0$. Moreover, we assume that $S(f) = 0$ if and only if $f = 0$. We will later refer to functionals of the above type as *nonlinear Rayleigh quotients*. It is easy to see that the standard Rayleigh quotient in (3.2) corresponding to the standard eigenvalue problem is a special case of the general functional in (3.3), if one restricts the matrix A to be positive semi-definite.

The differentiable case. To gain some intuition, let us first consider the case where R and S are differentiable. Then for every critical point f of Q,

$$\nabla Q(f) = 0 \qquad \Longleftrightarrow \qquad \nabla R(f) - \frac{R(f)}{S(f)} \cdot \nabla S(f) = 0 \;.$$

Let $r, s : \mathbb{R}^n \to \mathbb{R}^n$ be the operators defined as $r(f) = \nabla R(f)$, $s(f) = \nabla S(f)$ and $\lambda = \frac{R(f)}{S(f)}$, we see that every critical point f of Q is the solution to a problem of the form

$$r(f) - \lambda\, s(f) = 0. \tag{3.4}$$

We will later define this as *nonlinear eigenproblem*. Note that this is in general a system of nonlinear equations, as r and s are nonlinear operators. If R and S are both quadratic, r and s are linear operators, and hence (3.4) boils down to the standard eigenproblem (3.1).

The general non-differentiable case. Before we proceed to the general non-differentiable case, we have to introduce some important concepts from non-smooth analysis. Note that Q is in general non-convex and non-differentiable. For this reason we need to clarify what exactly one understands by the term *critical point* in the case of a non-convex, non-differentiable function. In the following we denote by $\partial_C Q(f)$ the *generalized gradient* of Q at f according to Clarke [1983],

$$\partial_C Q(f) = \{\xi \in \mathbb{R}^n \;\big|\; Q^0(f, v) \;\geq\; \langle \xi, v \rangle \,, \quad \text{for all } v \in \mathbb{R}^n\},$$

where $Q^0(f, v) \;=\; \lim_{g \to f,\, t \to 0} \sup \frac{Q(g+tv) - Q(g)}{t}$. The generalized gradient (also often referred to as Clarke subdifferential) generalizes the well-known subdifferential as in the case where Q is convex, $\partial_C Q$ is the subdifferential of Q and $Q^0(f, v)$ the directional derivative for each $v \in \mathbb{R}^n$. Moreover, one obtains the gradient if Q is differentiable. A characterization of critical points of non-smooth non-convex functionals is as follows, see Chang [1981].

Definition 3.5 (Critical point). *A point $f \in \mathbb{R}^n$ is called a critical point of Q if $0 \in \partial_C Q(f)$.*

This definition generalizes the well-known fact that the gradient of a differentiable function vanishes at a critical point. One can now show that a problem of the form (3.4) is a necessary condition for a critical point and in some cases even sufficient. The following theorem has been reported in Hein and Bühler [2010] for the special case where R and S are convex, i.e. $R_2(f) = 0$ and $S_2(f) = 0$, $\forall f \in \mathbb{R}^n$.

Theorem 3.6 (Critical points of nonlinear Rayleigh quotient). *Let the functions $R = R_1 - R_2$ and $S = S_1 - S_2$ fulfill the stated conditions. Then a necessary condition for $f \in \mathbb{R}^n$ being a critical point of Q is*

$$0 \in \partial R_1(f) - \partial R_2(f) - \lambda \left(\partial S_1(f) - S_2(f)\right), \tag{3.5}$$

where $\lambda = Q(f)$. If S, R_2 and S_2 are continuously differentiable at f, then this is also sufficient. Moreover, if (3.5) is fulfilled for some $\lambda \in \mathbb{R}$ and $f \in \mathbb{R}^n$, then $\lambda = Q(f)$.

Proof. Let f fulfill the general nonlinear eigenproblem in (3.5), where $r_1 \in \partial R_1(f), r_2 \in \partial R_2(f), s_1 \in \partial S_1(f)$ and $s_2 \in \partial S_2(f)$, such that $r_1 - r_2 - \lambda\,(s_1 - s_2) = 0$. Then by Lemma 2.8,

$$\begin{aligned} 0 &= \langle f, r_1\rangle - \langle f, r_2\rangle - \lambda\left(\langle f, s_1\rangle - \langle f, s_2\rangle\right) \\ &= p\,\left(R_1(f) - R_2(f)\right) - p\,\lambda\,\left(S_1(f) - S_2(f)\right), \end{aligned}$$

and thus $\lambda = R(f)/S(f)$. As R, S are Lipschitz continuous, one has, see Prop. 2.3.14 and 2.3.3 in Clarke [1983],

$$\begin{aligned} \partial_C\Big(\frac{R}{S}\Big)(f) &\subseteq \frac{S(f)\,\partial_C R(f) - R(f)\,\partial_C S(f)}{S(f)^2} && (3.6)\\ &\subseteq \frac{S(f)\,\left(\partial R_1(f) - \partial R_2(f)\right) - R(f)\,\left(\partial S_1(f) - \partial S_2(f)\right)}{S(f)^2} && (3.7)\\ &= \frac{1}{S(f)}\left(\partial R_1(f) - \partial R_2(f) - \frac{R(f)}{S(f)}\left(\partial S_1(f) - \partial S_2(f)\right)\right). \end{aligned}$$

Thus if f is a critical point, i.e. $0 \in \partial_C Q(f)$, then $0 \in \partial R_1(f) - \partial R_2(f) - \frac{R(f)}{S(f)}\left(\partial S_1(f) - \partial S_2(f)\right)$ given that $f \neq 0$. Moreover, by Prop. 2.3.14 and 2.3.3 in Clarke [1983] one has equality in (3.6) if S is continuously differentiable at f, and equality in (3.7) if R_2 and S_2 are continuously differentiable at f (see also Prop. 2.3.6). In this case, the fact that (3.5) is fulfilled implies that f is a critical point of Q. □

The results from the above theorem motivates the definition of nonlinear eigenproblems as follows:

Definition 3.7 (Nonlinear eigenproblem). *Let the functions $R = R_1 - R_2$ and $S = S_1 - S_2$ fulfill the stated conditions. Then the problem*

$$0 \in \partial R_1(f) - \partial R_2(f) - \lambda \left(\partial S_1(f) - \partial S_2(f)\right) \tag{3.8}$$

is called nonlinear eigenproblem. The solution f and λ are called nonlinear eigenvector and eigenvalue.

One easily checks that the above definition yields the standard linear eigenproblem as special case. Moreover, to see that the definition makes sense, let us state the following elementary property of nonlinear eigenproblems.

Proposition 3.8. *Let f be an eigenvector with eigenvalue λ according to the eigenproblem* (3.8)*. Then for any $\alpha \in \mathbb{R}$, the vector αf is an eigenvector with the same eigenvalue λ and it holds that $\lambda = Q(\alpha f)$.*

In the following, we use the notation $\phi : \mathbb{R} \to \mathbb{R}$, $\phi_p(x) = |x|^{p-2}\, x$.

Lemma 3.9. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be convex p-homogeneous for $p > 0$ and even. Moreover, let $r \in \partial R(f)$. Then for any $\alpha \in \mathbb{R}$ we have $\phi_p(\alpha)\, r \in \partial R(\alpha f)$.*

Proof. Let $r \in \partial R(f)$. Then $\forall g \in \mathbb{R}^n$ one has $R(g) \geq R(f) + \langle r, g - f\rangle$. Multiplying this by $|\alpha|^p$, one obtains $|\alpha|^p R(g) \geq |\alpha|^p R(f) + \left\langle |\alpha|^{p-2}\alpha\, r, \alpha g - \alpha f\right\rangle$. Using the p-homogeneity of R as well as the fact that it is even, one obtains $\forall g \in \mathbb{R}^n$, $R(\alpha g) \geq R(\alpha f) + \left\langle \phi_p(\alpha)\, r, \alpha g - \alpha f\right\rangle$. Substituting $h := \alpha g \in \mathbb{R}^n$ shows that $\phi_p(\alpha)\, r \in R(\alpha f)$. □

The proof of Prop. 3.8 is now straightforward.

Proof of Prop. 3.8. Let f be an eigenvector with eigenvalue λ according to the eigenproblem (3.8). Then there exist $r_1 \in \partial R_1(f)$, $r_2 \in \partial R_2(f)$, $s_1 \in \partial S_1(f)$ and $s_2 \in \partial S_2(f)$ such that $0 = r_1 - r_2 - \lambda(s_1 - s_2)$.

Multiplying this by $\phi_p(\alpha)$ we obtain $0 = \phi_p(\alpha)r_1 - \phi_p(\alpha)r_2 - \lambda(\phi_p(\alpha)s_1 - \phi_p(\alpha)s_2)$. Thus by Lemma 3.9 there exist $r_1' := \phi_p(\alpha)r_1 \in \partial R_1(\alpha f)$, $r_2' := \phi_p(\alpha)r_2 \in \partial R_2(\alpha f)$, $s_1' := \phi_p(\alpha)s_1 \in \partial S_1(\alpha f)$ and $s_2' := \phi_p(\alpha)s_2 \in \partial S_2(\alpha f)$ such that $0 = r_1' - r_2' - \lambda(s_1' - s_2')$, which implies that αf is an eigenvector with same eigenvalue λ. The last statement follows from Theorem 3.6. □

The generalization to non-quadratic functions enables us to consider a wider class of problems, and thus leads to a stronger modeling power. For example, Amghibech [2003, 2006] considered the nonlinear eigenproblem $\Delta_p(u) - \lambda\phi_p(u) = 0$, where Δ_p is the discrete graph p-Laplacian, a nonlinear operator which will be defined in Chapter 7. We will demonstrate the usefulness of this operator (especially in the limiting case $p = 1$) for the balanced graph cut problem. In Chapter 5 we present an efficient scheme to compute solutions of nonlinear eigenproblems.

Note that in the literature one finds a different class of problems also referred to as nonlinear eigenproblems. The eigenproblems defined in (3.8)

consist of systems which, seen as a function of the vector $f \in \mathbb{R}^n$, are in general nonlinear. However, seen as a function of λ, the right hand side of (3.8) is linear. A different type of nonlinear eigenproblem is given by a system where it is the other way round: here we have linearity in u but nonlinearity in λ, i.e. problems of the form $A(\lambda)f = 0$, where $A : \mathbb{R} \to \mathbb{R}^{n\times n}$ is a family of matrices depending on the variable of λ. These types of nonlinear eigenproblem have been intensely studied, see e.g. Mehrmann and Voss [2005]. In this work we restrict ourselves to problems of the form (3.8).

The question remains how to give a characterization of a sequence of eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots$. Note that the number of eigenvalues is in general larger than n and may even be infinite [Fučík et al., 1973]. We restrict ourselves now to the case where R and S are differentiable. Ljusternik and Schnirelmann derived a generalization of the Courant-Weinstein principle to nonlinear eigenproblems, see Fučík et al. [1973]. In order to transfer Theorem 3.4 to the nonlinear case, one needs to find a suitable generalization of the classes $\mathcal{U}_k$ (the set of k-dimensional subspaces). One possibility is by using the notion of the *Krasnoselskii genus* of a set, defined for a set $A \subset \mathbb{R}^n$ as [Drábek, 2012]

$$\gamma(A) := \inf\{m \in \mathbb{N} : \exists\, h : A \to \mathbb{R}^m\backslash\{0\},\ h \text{ is odd, i.e. } h(-x) = -h(x)\},$$

and $\gamma(A) := \infty$, if no such m exists. Intuitively, this means that the function h defines a mapping to an m-dimensional space where the elements of the set A are separated into positive and negative elements. The smallest value of m where this is possible can therefore be seen as a measure of the "dimension" of the set. Thus it makes sense to define the class $\mathcal{K}_k$ as the set of all closed symmetric subsets A of $\mathbb{R}^n$ with $\gamma(A) \geq k$. The following is the resulting characterization of a sequence of non-decreasing eigenvalues, see e.g. Fučík et al. [1973], Drábek [2012].

Theorem 3.10 (Ljusternik-Schnirelmann). *Let the functions R, S : $\mathbb{R}^n \to \mathbb{R}$ be p-homogeneous, even and differentiable. Moreover, let $\lambda_1 \leq \lambda_2 \leq \dots$ be a sequence of non-decreasing eigenvalues. Then one has,*

$$\lambda_k = \min_{K_k \in \mathcal{K}_k} \max_{\substack{u \in K_k \\ S(u)>0}} \frac{R(u)}{S(u)}.$$

Note that in general the above result does not cover all eigenvalues. Thus there exist alternative methods to construct sequences of eigenvalues, by using different choices of the classes $\mathcal{K}_k$, see Drábek [2012]. However, while these results are useful for the theoretical analysis of the spectrum of the nonlinear operators, up to our knowledge they all suffer from the same drawback that they do not yield a tractable technique for an explicit construction of higher order solutions of the nonlinear eigenproblems. For this reason, in the sequel we restrict ourselves to the computation of the smallest eigenvalues. An extension to higher eigenvalues will be the topic of future work.

Chapter 4

Tight relaxations of constrained fractional set programs

This chapter deals with a general class of constrained optimization problems involving ratios of set functions of the form

$$\begin{aligned} \min_{C \subset V} \frac{\widehat{R}(C)}{\widehat{S}(C)} &=: \widehat{Q}(C) \\ \text{subject to} : \widehat{M}_i(C) &\leq k_i, \quad i = 1, \ldots, K \end{aligned} \tag{4.1}$$

where $\widehat{R}, \widehat{S}, \widehat{M}_i : 2^V \to \mathbb{R}$ are set functions on a set $V = \{1, \ldots, n\}$. The above type of problems are referred to as *constrained fractional set programs* (CFSP). Problems of this form arise in many application such as balanced graph cuts or community detection, see the examples given in Section 1.1. We assume here that $\widehat{R}, \widehat{S}$ are non-negative and that $\widehat{R}(\emptyset) = \widehat{S}(\emptyset) = 0$. No assumptions are made on the set functions $\widehat{M}_i$, in particular they are not required to be non-negative. Thus also lower bound constraints can be written in the above form. Moreover, the formulation in (4.1) also encompasses the subset constraint $J \subset C$ (see (1.1) and (1.2)) as it can be written as equality constraint $|J| - |J \cap C| = 0$. Alternatively, in some cases a direct integration of the subset constraints into the objective is possible, as we will discuss in Chapters 8 and 9.

The goal of this chapter is to derive tight relaxations of the above optimization problems, i.e. show that for each problem of the form (4.1), there exist functions $R, S, T : \mathbb{R}^n \to \mathbb{R}$ such that the unconstrained problem

$$\min_{f \in \mathbb{R}^n_+} \frac{R(f) + T(f)}{S(f)}, \tag{4.2}$$

is *equivalent* to the problem (4.1) in the sense that the optimal values agree and the solution of the former problem can be transformed into a solution

of the latter, and vice versa. Going over to the Euclidean space will allow us derive an efficient scheme to compute solutions of (4.2) and hence also the original problem (4.1), which will be described in Chapter 5. Moreover, the form of the non-convex ratios in (4.2) allows us to relate the solution of the problem to the solution of a nonlinear eigenproblem (see Chapter 3).

4.1 Tight relaxation - The unconstrained case

Before considering tight relaxations of general constrained fractional set programs of the form given in (4.1), we first restrict ourselves to the case where we do not have any constraints, i.e. optimization problems of the form

$$\min_{C \subset V} \frac{\widehat{R}(C)}{\widehat{S}(C)} =: \widehat{Q}(C), \tag{4.3}$$

where $\widehat{R}, \widehat{S} : 2^V \to \mathbb{R}$ are set functions on a set $V = \{1, \ldots, n\}$. Moreover, we will also assume that $\widehat{R}(\emptyset) = \widehat{S}(\emptyset) = 0$ throughout this chapter. Given any continuous extensions R and S of $\widehat{R}$ and $\widehat{S}$, using the property that $R(\mathbf{1}_C) = \widehat{R}(C)$ for all $C \subset V$ (see Chapter 2), one can directly observe that the following continuous fractional program is a relaxation of problem (4.3):

$$\inf_{f \in \mathbb{R}^n_+} \frac{R(f)}{S(f)}.$$

In the following, we will show that if one chooses R, S to be the Lovász extensions R^L, S^L of $\widehat{R}, \widehat{S}$, respectively, the relaxation is in fact tight in the sense that the optimal values agree and the solution of the combinatorial problem can be computed from the solution of the problem on $\mathbb{R}^n_+$.

Given a vector $f \in \mathbb{R}^n$, one can construct a set C' by *optimal thresholding* of f: Defining the sets C_i as $C_i := \{j \in V | f_j \geq f_i\}$ for all $i = 1, \ldots, n$, one computes

$$C' = \operatorname*{arg\,min}_{C_i, i=1,\ldots,n} \frac{\widehat{R}(C_i)}{\widehat{S}(C_i)}.$$

The following lemma shows that optimal thresholding of a vector f always leads to non-increasing values of the objective.

Lemma 4.1. *Let $\widehat{R}, \widehat{S} : 2^V \to \mathbb{R}$ be non-negative set functions and $R^L, S^L : \mathbb{R}^n \to \mathbb{R}$ their Lovász extensions. Then for all $f \in \mathbb{R}^n_+$,*

$$\frac{R^L(f)}{S^L(f)} \geq \min_{i=1,\ldots,n} \frac{\widehat{R}(C_i)}{\widehat{S}(C_i)}.$$

Let furthermore $\widehat{R}(V) = \widehat{S}(V) = 0$, then the result holds for all $f \in \mathbb{R}^n$.

Proof. We assume wlog that the components of f are in increasing order $f_1 \leq f_2 \leq \cdots \leq f_n$. Using the definition of the Lovász extension, we obtain

$$\begin{aligned} R^L(f) &= \sum_{i=1}^{n-1} \widehat{R}(C_{i+1}) \left(f_{i+1} - f_i\right) + f_1 \widehat{R}(V) \\ &= \sum_{i=1}^{n-1} \frac{\widehat{R}(C_{i+1})}{\widehat{S}(C_{i+1})} \widehat{S}(C_{i+1}) \left(f_{i+1} - f_i\right) + \frac{\widehat{R}(V)}{\widehat{S}(V)} \widehat{S}(V) f_1 \\ &\geq \min_{j=1,\ldots,n} \frac{\widehat{R}(C_j)}{\widehat{S}(C_j)} \left(\sum_{i=1}^{n-1} \widehat{S}(C_{i+1}) \left(f_{i+1} - f_i\right) + f_1 \widehat{S}(V) \right) \\ &\geq \min_{j=1,\ldots,n} \frac{\widehat{R}(C_j)}{\widehat{S}(C_j)} \quad S^L(f), \end{aligned}$$

where we used the non-negativity of $\widehat{R}$ and $\widehat{S}$ as well as the fact that $f \in \mathbb{R}^n_+$. By assumption, $\widehat{S}$ is non-negative, which implies by Prop. 2.14 that also S^L is non-negative, and thus division by $S^L(f)$ gives the result. The second statement follows by noting that the terms $f_1 \widehat{R}(V)$ and $f_1 \widehat{S}(V)$ on the right side vanish for all $f \in \mathbb{R}$ if $\widehat{R}(V) = \widehat{S}(V) = 0$. □

The above lemma constitutes the main part in the proof of the following theorem, which shows that replacing the set functions $\widehat{R}$ and $\widehat{S}$ by their Lovász extensions R^L and S^L, respectively, leads to a tight relaxation in the sense that we obtain two equivalent problems.

Theorem 4.2 (Tight relaxation using Lovász extension). *Let $\widehat{R}, \widehat{S} : 2^V \to \mathbb{R}$ be non-negative set functions and $R^L, S^L : \mathbb{R}^n \to \mathbb{R}$ their Lovász extensions, respectively. Then, it holds that*

$$\min_{C \subset V} \frac{\widehat{R}(C)}{\widehat{S}(C)} \;=\; \min_{f \in \mathbb{R}^n_+} \frac{R^L(f)}{S^L(f)} \;.$$

Moreover, it holds for all $f \in \mathbb{R}^n_+$,

$$\frac{R^L(f)}{S^L(f)} \;\geq\; \min_{i=1,\ldots,n} \frac{\widehat{R}(C_i)}{\widehat{S}(C_i)}.$$

Thus a minimizer of the ratio of set functions can be found by optimal thresholding. Let furthermore $\widehat{R}(V) = \widehat{S}(V) = 0$, then all the above statements hold if one replaces $\mathbb{R}^n_+$ with $\mathbb{R}^n$.

In most cases, the above theorem is sufficient to derive a tight relaxation of a fractional set program. However, it is sometimes difficult to obtain closed forms of the Lovász extensions in practice, or handle them efficiently in the resulting optimization problem, see Chapter 5. For this reason, we

will derive a generalization of Theorem 4.2 which gives additional flexibility in obtaining the tight relaxation of the fractional set program. The proof of Theorem 4.2 will follow as a special case of the more general theorem.

The main fact used in generalizing Theorem 4.2 is the following: Given any 1-homogeneous convex extension R of a set function $\widehat{R}$, it can be upper bounded by the Lovász extension of $\widehat{R}$, i.e. the Lovász extension R^L is maximal over the set of all convex one-homogeneous extensions of $\widehat{R}$.

Lemma 4.3. *Let $\widehat{R} : 2^V \to \mathbb{R}$ be a set function with $\widehat{R}(\emptyset) = 0$. Let R^L be the Lovász extension of $\widehat{R}$, and R be any positively 1-homogeneous convex extension of $\widehat{R}$. Then, it holds $\forall f \in \mathbb{R}^n_+$ that*

$$R(f) \leq R^L(f).$$

If $R(\mathbf{1}) = R(-\mathbf{1}) = 0$, then the above inequality holds $\forall\, f \in \mathbb{R}^n$. Moreover, if for some $f \in \mathbb{R}^n$ it holds that $R(f) = R^L(f)$, then $\partial R(f) \subset \partial R^L(f)$.

Proof. Assume wlog that f is ordered in increasing order $f_1 \leq f_2 \leq \cdots \leq f_n$. By Lemma 2.9, it holds for every convex, positively 1-homogeneous function $R : \mathbb{R}^n \to \mathbb{R}$ that $R(f) \geq \langle u, f \rangle$, for all $f \in \mathbb{R}^n$ and $u \in U = \partial R(0)$. In particular, using the fact that R is an extension of $\widehat{R}$, one has

$$\widehat{R}(C_i) = R(\mathbf{1}_{C_i}) \geq \langle u, \mathbf{1}_{C_i} \rangle\,, \quad \forall i = 1, \ldots, n,$$

$\forall u \in U$. From this it follows that for all $f \in \mathbb{R}^n_+$,

$$\begin{aligned} R^L(f) &= \sum_{i=1}^{n-1} \widehat{R}(C_{i+1})\,(f_{i+1} - f_i) + f_1 \widehat{R}(V) \\ &\geq \sum_{i=1}^{n-1} \left\langle u, \mathbf{1}_{C_{i+1}} \right\rangle (f_{i+1} - f_i) + f_1 \langle u, \mathbf{1} \rangle = \sum_{i=1}^{n} f_i u_i, \end{aligned} \tag{4.4}$$

where we used that the terms $f_{i+1} - f_i$ are non-negative. As this holds for all $u \in U$ one obtains again with Lemma 2.9 for all $f \in \mathbb{R}^n_+$,

$$R^L(f) \geq \sup_{u \in U} \langle f, u \rangle = R(f)\ .$$

For the second statement, let now $R(\mathbf{1}) = R(-\mathbf{1}) = 0$. On the one hand one has $0 = R(\mathbf{1}) \geq \langle u, \mathbf{1} \rangle$ for all $u \in U$. On the other hand, one has $0 = R(-\mathbf{1}) \geq \langle u, -\mathbf{1} \rangle$ for all $u \in U$. Thus it must hold that $\langle u, \mathbf{1} \rangle = 0$ for all $u \in U$, which implies that the lower bound in (4.4) holds for all $f \in \mathbb{R}^n$. For the last statement, let $r \in \partial R(f)$. Then $\forall g \in \mathbb{R}^n$,

$$R^L(f) + \langle r, g - f \rangle = R(f) + \langle r, g - f \rangle \leq R(g) \leq R^L(g),$$

which implies that $r \in \partial R^L(f)$ and therefore $\partial R(f) \subset \partial R^L(f)$. □

The property in Lemma 4.3 can now be used to show that given a decomposition of $\widehat{R}$ and $\widehat{S}$ into a difference of (submodular) set functions, one needs the Lovász extension only for the first term of $\widehat{R}$ and the second term of $\widehat{S}$. The remaining terms can be replaced by any convex 1-homogeneous extensions of the corresponding set functions. Note that by Proposition 2.21 such a decomposition always exists. The following theorem states that this leads to a tight relaxation of the fractional set program.

Theorem 4.4 (Tight relaxation - General version). *Let $\widehat{R}, \widehat{S} : 2^V \to \mathbb{R}$ be non-negative set functions and $\widehat{R} := \widehat{R}_1 - \widehat{R}_2$ and $\widehat{S} := \widehat{S}_1 - \widehat{S}_2$ be decompositions into differences of set functions. Let the Lovász extensions of $\widehat{R}_1, \widehat{S}_2$ be given by R_1^L, S_2^L and let R_2, S_1 be positively 1-homogeneous convex extensions of $\widehat{R}_2, \widehat{S}_1$ such that $S_1 - S_2^L$ is non-negative. Then,*

$$\min_{C \subset V} \frac{\widehat{R}(C)}{\widehat{S}(C)} = \min_{f \in \mathbb{R}^n_+} \frac{R_1^L(f) - R_2(f)}{S_1(f) - S_2^L(f)} .$$

Moreover, it holds for all $f \in \mathbb{R}^n_+$,

$$\frac{R_1^L(f) - R_2(f)}{S_1(f) - S_2^L(f)} \geq \min_{i=1,\dots,n} \frac{\widehat{R}(C_i)}{\widehat{S}(C_i)}.$$

Thus a minimizer of the ratio of set functions can be found by optimal thresholding. Let furthermore $\widehat{R}(V) = \widehat{S}(V) = 0$ and $R_2(\alpha \mathbf{1}) = S_1(\alpha \mathbf{1}) = 0$ for all $\alpha \in \{-1, 1\}$, then all the above statements hold if one replaces $\mathbb{R}^n_+$ with $\mathbb{R}^n$.

Proof. Let $R := R_1^L - R_2$ and $S := S_1 - S_2^L$. Moreover, let R_2^L and S_1^L be the Lovász extensions of $\widehat{R}_2$ and $\widehat{S}_1$. With Lemma 4.3, we get $\forall f \in \mathbb{R}^n_+$, $R(f) = R_1^L(f) - R_2(f) \geq R_1^L(f) - R_2^L(f) = R^L(f)$, and $S(f) = S_1(f) - S_2^L(f) \leq S_1^L(f) - S_2^L(f) = S^L(f)$. Thus, for all $f \in \mathbb{R}^n_+$,

$$\frac{R(f)}{S(f)} \geq \frac{R^L(f)}{S^L(f)}.$$

Moreover, if $R_2(\alpha \mathbf{1}) = S_1(\alpha \mathbf{1}) = 0$ for all $\alpha \in \{-1, 1\}$, the previous statements hold for all $f \in \mathbb{R}^n$. Applying Lemma 4.1 then yields

$$\frac{R^L(f)}{S^L(f)} \geq \min_{i=1,\dots,n} \frac{\widehat{R}(C_i)}{\widehat{S}(C_i)}$$

for all $f \in \mathbb{R}^n_+$, or for all $f \in \mathbb{R}^n$ if $\widehat{R}(V) = \widehat{S}(V) = 0$. Thus we obtain

$$\inf_{f \in \mathbb{R}^n_+} \frac{R(f)}{S(f)} \geq \inf_{f \in \mathbb{R}^n_+} \min_{\substack{C_i \text{ def. by } f \\ i=1,\dots,n}} \frac{\widehat{R}(C_i)}{\widehat{S}(C_i)} \geq \min_{A \subset V} \frac{\widehat{R}(A)}{\widehat{S}(A)} , \tag{4.5}$$

and the analogous result for $f \in \mathbb{R}^n$ if $\widehat{R}(V) = \widehat{S}(V) = 0$ and $R_2(\alpha \mathbf{1}) = S_1(\alpha \mathbf{1}) = 0$. On the other hand, using that R and S are extensions of $\widehat{R}$ and $\widehat{S}$, respectively, one has

$$\min_{A \subset V} \frac{\widehat{R}(A)}{\widehat{S}(A)} = \min_{A \subset V} \frac{R(\mathbf{1}_A)}{S(\mathbf{1}_A)} \geq \inf_{f \in \mathbb{R}^n_+} \frac{R(f)}{S(f)} \geq \inf_{f \in \mathbb{R}^n} \frac{R(f)}{S(f)} .$$

Combining this with (4.5), and using the fact that the infimum is achieved for some $\mathbf{1}_A \in \mathbb{R}^n_+$, one obtains the result. □

Note that no assumptions except non-negativity are made on $\widehat{R}$ and $\widehat{S}$ - *every* non-negative fractional set program has a tight relaxation into a continuous fractional program. The above theorem yields Theorem 4.2 as special case if one chooses R_2 and S_1 to be the Lovász extensions of $\widehat{R}_2$ and $\widehat{S}_1$, respectively. Note that if the condition $\widehat{R}(V) = 0$ holds, we can wlog assume that $\widehat{R}_1(V) = \widehat{R}_2(V) = 0$, and thus $R_2^L(\alpha \mathbf{1}) = \alpha \widehat{R}_2(V) = 0$, and similarly for $\widehat{S}$.

Compared to the first version of this theorem in [Bühler et al., 2013], the set functions $\widehat{R}_1, \widehat{R}_2$ and $\widehat{S}_1, \widehat{S}_2$ in the decompositions of $\widehat{R}$ and $\widehat{S}$ do not need to be submodular, as long as the extensions R_1, R_2, S_1, S_2 fulfill the stated conditions. However, in practice one would choose such a decomposition as the resulting tight relaxation will be a ratio of differences of convex functions which can be optimized using RatioDCA, see Chapter 5.

A special case of Theorem 4.4 was considered in Hein and Setzer [2011]. They treated the case where $\widehat{S}$ is symmetric, i.e. $\widehat{S}(C) = \widehat{S}(\overline{C})$, and $\widehat{R}$ is a cut function, which is submodular and symmetric. Then it was shown that

$$\min_{f \in \mathbb{R}^n} \frac{R^L(f)}{S(f)} = \min_{C \subset V} \frac{\widehat{R}(C)}{\widehat{S}(C)},$$

under two different conditions on the function S:

- S is the Lovász extension of the set function $\widehat{S}$.
- S is a convex, 1-homogeneous extension of $\widehat{S}$ which is even (i.e. $S(f) = S(-f), \forall f \in \mathbb{R}^n$) and satisfies $S(f + \alpha \mathbf{1}) = S(f), \forall f \in \mathbb{R}^n, \alpha \in \mathbb{R}$.

The first case is recovered from Theorem 4.4 by choosing S_1 as the Lovász extension of $\widehat{S}_1$. The second case is recovered by setting $\widehat{S}_2(C) = 0$ for all $C \subset V$. To see why in both cases $\mathbb{R}^n_+$ can be replaced by $\mathbb{R}^n$, note that due to the symmetry of $\widehat{S}$, we have $\widehat{S}(V) = 0$. Thus, in the first case, wlog one can assume that $\widehat{S}_1(V) = \widehat{S}_2(V) = 0$, which implies that $S_1^L(\alpha \mathbf{1}) = \alpha \widehat{S}_1(V) = 0$, for $\alpha \in \{-1, 1\}$. In the second case, the condition $S_1(f) = S_1(f + \alpha \mathbf{1})$ implies that $S_1(\alpha \mathbf{1}) = \widehat{S}_1(V) = \widehat{S}_1(\emptyset) = 0$ for $\alpha \in \{-1, 1\}$. The condition that S is even is not necessary to derive a tight relaxation. Moreover, due to the symmetry of $\widehat{R}$, we must have $\widehat{R}(V) = 0$, which implies that $R^L(\alpha \mathbf{1}) = 0$.

As observed by Jost et al. [2013], the results of Lemma 4.3 have important practical consequences for the development of an algorithm on $\mathbb{R}^n$ to

solve the constrained fractional set program. As we will discuss in Section 5.4.4, the RatioDCA introduced in the next chapter will achieve better results when applied to a tight relaxation based on Lovász extensions. Therefore, if the Lovász extension can be handled easily in the optimization, it is in fact the best choice. It turns out that the RatioDCA can be implemented without a closed form of the Lovász extension as only its subgradient is necessary which can always be computed. As we will see in Section 6.5, this approach is useful if the original set functions can be computed efficiently.

4.2 Tight relaxation - The constrained case

To solve the constrained fractional set program (4.1) we make use of the concept of *exact penalization* [Di Pillo, 1994], where the main idea is to transform a given constrained optimization problem into an *equivalent* unconstrained one by adding a suitable penalty term. The penalty term has to be chosen in such a way that the optimal solution of the unconstrained problem is also an optimal (and in particular feasible) solution for the constrained problem. We use this idea for our constrained fractional set programs and define for each constraint $\widehat{M}_i(C) \leq k_i$ a penalty set function as

$$\widehat{T}_i(C) = \begin{cases} \max\left\{0, \widehat{M}_i(C) - k_i\right\}, & C \neq \emptyset, \\ 0, & C = \emptyset. \end{cases} \tag{4.6}$$

The function $\widehat{T}_i(C)$ is chosen in such a way that it is zero if C satisfies the i-th constraint. Otherwise it attains a positive value which increases with "increasing infeasibility", i.e. when C has an increasing distance from the boundary of the constraint set. The special treatment of the empty set in the definition of $\widehat{T}_i$ is a technicality required for the derivation of the Lovász extension. In the following, the constant θ_i quantifies a "minimum value" of $\widehat{T}_i$ on the infeasible sets:

$$\theta_i = \min_{i=1,\ldots,K} \big[\min_{\widehat{M}_i(C) > k_i} \widehat{M}_i(C) - k_i \big]. \tag{4.7}$$

By construction, we have $\theta_i > 0$ (assuming that there exists at least one set $C \neq 0$ which is infeasible). For example, if $\widehat{M}_i(C) = |C|$ and k_i is some natural number, then θ_i is equal to 1. If $\widehat{M}_i(C) = \mathrm{vol}_g(C)$ and all vertex weights $g \in \mathbb{R}^n_+$ are rational numbers which are multiples of a fraction $\frac{1}{\rho}, \rho \in \mathbb{N}$, then $\theta_i \geq \frac{1}{\rho}$. The total penalty term is then given as

$$\widehat{T}_\gamma(C) := \sum_{i=1}^{K} \gamma_i \, \widehat{T}_i(C), \tag{4.8}$$

where $\gamma \in \mathbb{R}_+^K$ is a vector of non-negative parameters. Adding the above penalty term to the numerator of the objective yields the modified problem

$$\min_{C \subset V} \frac{\widehat{R}(C) + \widehat{T}_\gamma(C)}{\widehat{S}(C)} =: \widehat{Q}_\gamma(C). \tag{4.9}$$

We will show that using a feasible set of (4.1) one can compute a vector γ such that (4.9) is equivalent to the original constrained problem. Once we have established the equivalence, we can then apply Theorem 4.2 (note that $\widehat{T}_\gamma$ is a non-negative set function). This leads to the main result of this chapter, which states that there exists a tight relaxation of *all* problems of the form (4.1) where $\widehat{R}, \widehat{S}$ are non-negative set functions.

We first show the equivalence between the constrained problem (4.1) and the unconstrained problem (4.9) for the given choice of γ.

Lemma 4.5. *Let $\widehat{R}, \widehat{S} : 2^V \to \mathbb{R}$ be non-negative set functions. Let $\widehat{T}_\gamma$ by defined as in* (4.8)*. Then,*

$$\min_{\substack{\widehat{M}_i(C) \leq k_i, \\ i=1,\dots,K}} \frac{\widehat{R}(C)}{\widehat{S}(C)} = \min_{C \subset V} \frac{\widehat{R}(C) + \widehat{T}_\gamma(C)}{\widehat{S}(C)},$$

for any $\gamma \in \mathbb{R}_+^K$ such that $\forall i, \gamma_i > \frac{\widehat{R}(C_0)}{\theta_i \widehat{S}(C_0)} \max_{C \subset V} \widehat{S}(C)$, where $C_0 \subset V$ is a feasible set with $\widehat{S}(C_0) > 0$.

Proof. Note that for any feasible subset, i.e. a set C satisfying $\widehat{M}_i(C) \leq k_i$, $i = 1, \dots, K$, the objective $\widehat{Q}_\gamma$ of problem (4.9) is equal to the objective $\widehat{Q}$ of problem (4.1). Thus, if we show that *all* minimizers of the second problem satisfy the constraints, the equivalence follows.

Suppose that $C^* \neq \emptyset$ is a minimizer of the second problem and that C^* is infeasible. Without loss of generality, assume that the first K_1 constraints are violated, where $1 \leq K_1 \leq K$. Then by definition we have $\widehat{T}_i(C^*) \geq \theta_i$ for all $i \leq K_1$, and $\widehat{T}_i(C^*) = 0$ for $i > K_1$. This yields

$$\begin{aligned} \widehat{Q}_\gamma(C^*) &= \frac{\widehat{R}(C^*) + \sum_{i=1}^K \gamma_i \widehat{T}_i(C^*)}{\widehat{S}(C^*)} \\ &\geq \frac{\widehat{R}(C^*) + \sum_{i=1}^{K_1} \gamma_i \theta_i}{\widehat{S}(C^*)} \geq \frac{\sum_{i=1}^{K_1} \gamma_i \theta_i}{\widehat{S}(C^*)} \geq \frac{\sum_{i=1}^{K_1} \gamma_i \theta_i}{\max_{C \subset V} \widehat{S}(C)}, \end{aligned} \tag{4.10}$$

where we used the non-negativity of $\widehat{R}$ and $\widehat{S}$. Hence by the condition on γ,

$$\widehat{Q}_\gamma(C^*) > K_1 \frac{\widehat{R}(C_0)}{\widehat{S}(C_0)} \geq \frac{\widehat{R}(C_0)}{\widehat{S}(C_0)} = \widehat{Q}(C_0) = \widehat{Q}_\gamma(C_0), \tag{4.11}$$

which contradicts the fact that C^* is optimal. □

Note that in practice, the value of the constants θ_i as well as the bounds on the parameters γ_i are never explicitly computed. Instead we start with the unconstrained case $\gamma = 0$, and then increase γ sequentially until all constraints are fulfilled (see experimental section). Moreover, when having multiple penalty terms, in practice it also makes sense to rescale the penalty terms such that they achieve values in the same range, see Section 8.4.2. We can now use the above lemma to derive the following result which shows that tight relaxations exists for all constrained fractional set programs.

Theorem 4.6 (Tight relaxation using Lovász extension). *Let $\widehat{R}, \widehat{S} : 2^V \to \mathbb{R}$ be non-negative set functions and R^L, S^L their Lovász extensions. Denote by T^L_γ the Lovász extension of the function $\widehat{T}_\gamma$ defined in* (4.8)*. Then,*

$$\min_{\substack{\widehat{M}_i(C) \leq k_i, \\ i=1,\dots,K}} \frac{\widehat{R}^L(C)}{\widehat{S}^L(C)} \;=\; \min_{f \in \mathbb{R}^n_+} \frac{R^L(f) + T^L_\gamma(f)}{S^L(f)} \;:=\; Q_\gamma(f)$$

for any $\gamma \in \mathbb{R}^K_+$ such that $\forall i, \gamma_i > \frac{\widehat{R}(C_0)}{\theta_i \, \widehat{S}(C_0)} \max_{C \subset V} \widehat{S}(C)$, where $C_0 \subset V$ is a feasible set with $\widehat{S}(C_0) > 0$. Moreover, for any $f \in \mathbb{R}^n_+$ with $Q_\gamma(f) < \widehat{Q}_\gamma(C_0)$ for the given γ, we have

$$Q_\gamma(f) \;\geq\; \min_{i=1,\dots,n} \widehat{Q}_\gamma(C_i),$$

and the minimizing set on the right hand side is feasible. Let furthermore $\widehat{R}(V) = \widehat{S}(V) = \widehat{T}_\gamma(V) = 0$, then all the above statements hold if one replaces $\mathbb{R}^n_+$ with $\mathbb{R}^n$.

Similarly to the unconstrained case, one can derive a generalized version of Theorem 4.6, where for decompositions of the numerator and denominator into differences of set functions, only the first part of the numerator and second part of the denominator are replaced by their Lovász extensions, while for the other parts any convex non-negative 1-homogeneous extension can be used. Analogously to before, the proof of Theorem 4.6 will then follow as special case of the more general theorem.

Theorem 4.7 (Tight relaxation - General version). *Let $\widehat{R}, \widehat{S} : 2^V \to \mathbb{R}$ be non-negative set functions and $\widehat{T}_\gamma$ be defined as in* (4.8)*. Moreover, let $\widehat{R} = \widehat{R}_1 - \widehat{R}_2$, $\widehat{S} = \widehat{S}_1 - \widehat{S}_2$ and $\widehat{T}_\gamma = \widehat{T}_1 - \widehat{T}_2$ be decompositions into differences of set functions. Let the Lovász extensions of $\widehat{R}_1, \widehat{S}_2, \widehat{T}_1$ be given by R^L_1, S^L_2, T^L_1 and let R_2, S_1, T_2 be positively 1-homogeneous convex extensions of $\widehat{R}_2, \widehat{S}_1, \widehat{T}_2$ such that $S_1 - S^L_2$ is non-negative. Then,*

$$\min_{\substack{\widehat{M}_i(C) \leq k_i, \\ i=1,\dots,K}} \frac{\widehat{R}(C)}{\widehat{S}(C)} \;=\; \min_{f \in \mathbb{R}^n_+} \frac{R^L_1(f) - R_2(f) + T^L_1(f) - T_2(f)}{S_1(f) - S^L_2(f)} \;:=\; Q_\gamma(f)$$

for any $\gamma \in \mathbb{R}^K_+$ *such that* $\forall i, \gamma_i > \frac{\widehat{R}(C_0)}{\theta_i \widehat{S}(C_0)} \max_{C \subset V} \widehat{S}(C)$, *where* $C_0 \subset V$ *is a feasible set with* $\widehat{S}(C_0) > 0$. *Moreover, for any* $f \in \mathbb{R}^n_+$ *with* $Q_\gamma(f) < \widehat{Q}_\gamma(C_0)$ *for the given* γ, *we have*

$$Q_\gamma(f) \;\geq\; \min_{i=1,\ldots,n} \widehat{Q}_\gamma(C_i),$$

and the minimizing set on the right hand side is feasible. Let furthermore $\widehat{R}(V) = \widehat{S}(V) = \widehat{T}_\gamma(V) = 0$ *and* $R_2(\alpha \mathbf{1}) = S_1(\alpha \mathbf{1}) = T_2(\alpha \mathbf{1}) = 0$ *for all* $\alpha \in \{-1, 1\}$, *then all the above statements hold if one replaces* $\mathbb{R}^n_+$ *with* $\mathbb{R}^n$.

Proof. By Lemma 4.5, one has for the given choice of γ,

$$\min_{\substack{\widehat{M}_i(C) \leq k_i, \\ i=1,\ldots,K}} \frac{\widehat{R}(C)}{\widehat{S}(C)} \;=\; \min_{C \subset V} \frac{\widehat{R}(C) + \widehat{T}_\gamma(C)}{\widehat{S}(C)}.$$

Noting that $\widehat{T}_\gamma$ is a non-negative set function with $\widehat{T}_\gamma(\emptyset) = 0$ and $\gamma_i > 0$, we have a ratio of non-negative set functions which attain the value zero on the empty set. Writing the terms $\widehat{R}, \widehat{S}$ and $\widehat{T}_\gamma$ as differences of submodular functions (by Proposition 2.21 such a decomposition always exists) and then applying Theorem 4.4 yields the equivalent continuous problem. Moreover, if $\widehat{R}(V) = \widehat{S}(V) = \widehat{T}_\gamma(V) = 0$ and $R_2(\alpha \mathbf{1}) = S_1(\alpha \mathbf{1}) = T_2(\alpha \mathbf{1}) = 0$, we also have $(\widehat{R} + \widehat{T}_\gamma)(V) = 0$ and $(R_2 + T_2)(\alpha \mathbf{1}) = 0$, thus the conditions in Theorem 4.4 are fulfilled and we can replace $\mathbb{R}^n_+$ by $\mathbb{R}^n$.

The second statement can be seen as follows. Suppose $Q_\gamma(f) < \widehat{Q}_\gamma(C_0)$. By Lemma 4.1 and Lemma 4.3 we obtain

$$Q_\gamma(f) \;\geq\; \min_{i=1,\ldots,n} \widehat{Q}_\gamma(C_i).$$

Now suppose that the minimizer C^* of the right hand side is not feasible, then again by the derivation in (4.10) and (4.11) and the choice of γ,

$$\widehat{Q}_\gamma(C^*) \;>\; \widehat{Q}_\gamma(C_0),$$

which leads to a contradiction. Thus C^* is feasible. □

Note that Theorem 4.7 implies that the set found by optimal thresholding of the solution of the continuous program is guaranteed to satisfy all constraints. In Chapter 8 and 9 we will derive tight relaxations for the constrained local clustering problem and the constrained densest subgraph problem introduced in (1.1) and (1.2), where we can guarantee that all constraints are fulfilled. We are not aware of any other method which can give the same guarantee for these problems. In the next part of the thesis, we discuss algorithms for the resulting optimization problems.

Part II

Algorithms for fractional programs

Chapter 5

Optimization of ratios of non-negative functions

This chapter is concerned with the algorithmic solution of a class of optimization problems of the form

$$\min_{f \in \mathbb{R}^n} \frac{R(f)}{S(f)} := Q(f), \tag{5.1}$$

for some non-negative functions $R, S : \mathbb{R}^n \to \mathbb{R}_+$. Moreover, we also consider the case where the optimization is done over the positive orthant $\mathbb{R}^n_+$. Optimization problems of this type arise as the result of the tight relaxation of fractional set programs, see Chapter 4. Furthermore, many other problems can be directly modeled in this form, which makes the considered algorithms widely applicable. Moreover, in Chapter 3 we have shown the connection of the critical points of above functionals to solutions of the associated nonlinear eigenproblems. The algorithms presented in this chapter will be used in Chapters 7 to 10 to develop techniques for several applications in network analysis and dimensionality reduction.

We start our discussion by considering an important special case: in the first section, we cover the case when R is the quadratic form $\langle f, Af \rangle$ induced by a (symmetric) positive semi-definite matrix $A \in \mathbb{R}^{n \times n}$, and S is the squared Euclidean norm. In this case, the problem can be understood as computing the smallest eigenvalue of the matrix A, which can be solved using the well-known inverse power method (see e.g. Golub and Van Loan [1996]). In Section 5.2 we assume that R is convex and S is concave. Under some further assumptions on R and S, a variant of the above problem can then be solved globally optimally by Dinkelbach's method [Dinkelbach, 1967]. Section 5.3 covers the case where R and S are non-negative convex p-homogeneous functions for $p \geq 1$. We derive a nonlinear inverse power method which generalizes the standard inverse power method, and can be shown to converge to a solution of the associated nonlinear eigenproblem (see Chapter 3).

The case $p = 1$ has been further generalized by Hein and Setzer [2011], Jost et al. [2013] to arbitrary ratios of non-negative differences of convex 1-homogeneous functions, which is discussed in Section 5.4. We then present a variant of the RatioDCA of Hein and Setzer [2011] extended to arbitrary ratios of non-negative differences of convex functions. If R and S are p-homogeneous ($p \geq 1$), it converges to the solution of a nonlinear eigenproblem, see Chapter 3.

5.1 Quadratic function over quadratic function: Standard inverse power method

In this section, we briefly discuss the special case of problem (5.1) where $R(f) = \langle f, Af \rangle$ for a positive semi-definite matrix $A \in \mathbb{R}^{n \times n}$, and $S(f) = \|f\|_2^2$. The problem (5.1) then has the form

$$\min_{f \in \mathbb{R}^n} \frac{\langle f, Af \rangle}{\|f\|_2^2} =: \min_{f \in \mathbb{R}^n} Q(f). \tag{5.2}$$

By the Rayleigh-Ritz principle (see Chapter 3), the minimum of the above problem is equal to the smallest eigenvalue of the matrix A, and the minimizer is the corresponding eigenvector.

The *power method* is a standard technique to compute the dominant eigenvalue (the one with largest absolute value) of a symmetric matrix A (see e.g. Golub and Van Loan [1996]). Its main building block is the fact that the iterative scheme

$$f^{k+1} = Af^k \tag{5.3}$$

converges to the dominant eigenvalue of A, which can be easily seen by expressing the initial vector f^0 in terms of the orthonormal basis of eigenvectors $u_1, \ldots, u_n$, and then analyzing the dominant terms in the sum as the iterative scheme progresses. Since we assume the matrix A to be positive semi-definite, all eigenvalues are non-negative, and therefore one obtains the largest eigenvalue of A.

Now consider the matrix $A' = (A - \mu I)^{-1}$ (assuming A is invertible). One easily shows that for each eigenpair (λ, u) of A, the pair $((\lambda - \mu)^{-1}, u)$ is an eigenpair of A'. Hence, given the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A, the largest eigenvalue of A' corresponds to the smallest value of $\lambda_i - \mu$, which is achieved by the eigenvalue of A closest to μ. This implies that applying the iterative scheme (5.3) to A' yields the eigenvalue of A closest to μ. This idea is used in the *inverse power method* summarized in Alg. 1.

Applying the inverse power method for $\mu = 0$ then converges to the eigenvector of A corresponding to the smallest eigenvalue, which equals the minimum of the functional in (5.2). Note that since A is positive semi-definite, it holds that (5.2) is a ratio of two convex functions. In Section

Algorithm 1 Standard inverse power method

1: **Input:** eigenvalue estimate $\mu \in \mathbb{R}$
2: **Initialization:** $f^0 = \text{random with } \left\|f^0\right\|_2 = 1,\ \lambda^0 = Q(f^0)$
3: **repeat**
4: Solve $(A - \mu I)g^{k+1} = f^k$ for g^{k+1}
5: $f^{k+1} = g^{k+1} / \left\|g^{k+1}\right\|_2$
6: $\lambda^{k+1} = Q(f^{k+1})$
7: **until** $\frac{\left|\lambda^{k+1} - \lambda^k\right|}{\lambda^k} < \epsilon$
8: **Output:** eigenvalue λ^{k+1} closest to μ with eigenvector f^{k+1}.

5.3 we will generalize the inverse power method to nonlinear eigenproblems of the form $0 \in \partial R(u) - \lambda\, \partial S(u)$, where R and S are non-negative, convex, p-homogeneous functions ($p \geq 1$).

5.2 Convex function over concave function: Dinkelbach's method

We now consider the case where the functional R is convex and S is concave. There has been a large body of work on convex-concave fractional programs and also the special class of linear fractional programs. Examples of this type of problems include resource allocation problems, where certain ratios between cost and return are optimized, the portfolio selection problem, where we want to minimize risk while maximizing return, or problems where the cost per time needs to be minimized (see Schaible [1981]).

Since it does not make sense to restrict ourselves only to concave functions which are non-negative on $\mathbb{R}^n$, we consider a slightly modified problem to before. The considered optimization problem is given by

$$\min_{f \in C} \frac{R(f)}{S(f)}, \tag{5.4}$$

where $C \subset \mathbb{R}^n$ is a compact convex subset of $\mathbb{R}^n$, and $R, S : \mathbb{R}^n \to \mathbb{R}$ are assumed to be non-negative inside the set C. Moreover, we assume R to be convex and S to be concave.

Proposition 5.1. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be convex and $S : \mathbb{R}^n \to \mathbb{R}$ be concave, and let $R(f) \geq 0$ and $S(f) \geq 0$ for all $f \in C$. Then the function $Q : C \to \mathbb{R}_+, Q(f) := \frac{R(f)}{S(f)}$ is a quasi-convex function.*

Proof. For any $\alpha \in \mathbb{R}$, the sublevel set C_α is given as

$$C_\alpha = \{f \in C \mid \tfrac{R}{S}(f) \leq \alpha\} = \{f \in C \mid R(f) - \alpha S(f) \leq 0\}.$$

If $\alpha < 0$, we have $C_\alpha = \emptyset$ due to the non-negativity of R and S. Otherwise, the function $R(f) - \alpha S(f)$ is a convex function, and hence C_α is a convex set $\forall \alpha \in \mathbb{R}$, which implies that $\frac{R}{S}$ is a quasi-convex function. □

The above result implies that the ratio problem can be solved globally optimally, see Boyd and Vandenberghe [2004]. In particular, the form of the sublevel sets C_α suggests the following iterative procedure to compute the global minimum of the functional Q. Assume we know an upper bound α on the optimal value of Q, i.e. $\min_{u \in C} Q(u) \leq \alpha$. Then, a vector $f \in \mathbb{R}^n$ satisfying $R(f) - \alpha S(f) \leq 0$ implies that $f \in C_\alpha$ and hence $\min_{u \in C} Q(u) \leq Q(f) \leq \alpha$. If $Q(f)$ is strictly smaller than α, we have obtained a better upper bound on the optimal value of Q. This suggests to solve the problem via a sequence of improving upper bounds on the objective, which directly leads to the method of Dinkelbach [1967] shown in Alg. 2.

Algorithm 2 Dinkelbach's method for convex over concave

1: **Initialization:** $f^0 \in C$, $\lambda^0 = Q(f^0)$
2: **repeat**
3: $\quad f^{k+1} = \arg\min_{u \in C} \left\{ R(u) - \lambda^k S(u) \right\}$
4: $\quad \lambda^{k+1} = R(f^{k+1}) / S(f^{k+1})$
5: **until** $\frac{|\lambda^{k+1} - \lambda^k|}{\lambda^k} < \epsilon$
6: **Output:** global minimum λ^{k+1} with minimizer f^{k+1}.

Note that in the original paper [Dinkelbach, 1967], the problem is formulated as a maximization problem. Moreover, a slightly different stopping criterion is used. In the following, we adapt the convergence results in [Dinkelbach, 1967] for the variant given in Alg. (2). Let $\Phi_{f^k}(u) := R(u) - \lambda^k S(u)$ denote the objective of the convex inner problem in Alg. (2) at step k.

Lemma 5.2. *The sequence f^k generated by Alg. 2 satisfies $Q(f^k) > Q(f^{k+1})$ for all $k \geq 0$ or terminates.*

Proof. By assumption it holds that $\lambda^k = Q(f^k) \geq 0$. Thus, since R is convex and S is concave, it must hold that Φ_{f^k} is convex. Moreover, note that we have $\Phi_{f^k}(f^k) = R(f^k) - \lambda^k S(f^k) = 0$, thus the optimizer f^{k+1} of the inner problem satisfies $R(f^{k+1}) - \lambda^k S(f^{k+1}) \leq 0$. If equality holds, then f^k is a possible minimizer and the sequence terminates. Otherwise, we have $R(f^{k+1}) - \lambda^k S(f^{k+1}) < 0$. Assume now that $S(f^{k+1}) = 0$. This implies that $R(f^{k+1}) < 0$, contradicting the fact that R is non-negative. Thus, $S(f^{k+1}) > 0$ and one obtains

$$\frac{R(f^{k+1})}{S(f^{k+1})} < \lambda_k = \frac{R(f^k)}{S(f^k)},$$

which concludes the proof. □

Lemma 5.3. *The sequence f^k produced by Alg. 2 satisfies $\lim_{k\to\infty} Q(f^k) = \lambda^*$, where λ^* is the global minimum of the functional Q.*

Proof. By Lemma 5.2 the sequence $Q(f^k)$ is monotonically decreasing. By assumption R and S are non-negative in C. Thus Q is bounded below by zero, which implies convergence towards a limit

$$\lambda^* = \lim_{k\to\infty} Q(f^k) \ .$$

The fact that the sequences f^k are contained in the compact set C implies the existence of a subsequence f^{k_j} converging to some element $f^* \in C$. As the sequence $Q(f^{k_j})$ is a subsequence of a convergent sequence, it has to converge towards the same limit λ^*. Let $Q(f')$ denote the global optimal solution of problem (5.4). Assume that $\lambda^* = Q(f^*) > Q(f')$. This implies (note that $S(f') > 0$)

$$R(f') - \frac{R(f^*)}{S(f^*)} S(f') < 0 = R(f^*) - \frac{R(f^*)}{S(f^*)} S(f^*).$$

Thus, $\Phi_{f^*}(f') < \Phi_{f^*}(f^*)$, which contradicts the fact that f^* is the optimal solution of the inner problem (note that $f', f^* \in C$). Thus our assumption is wrong and we must have $Q(f^*) = Q(f')$. □

Note that Dinkelbach's method cannot be applied to the case where R and S are both convex, as in this case the problem $R(u) - \lambda^k S(u)$ is not guaranteed to be convex. For this reason, in the next section, we present our nonlinear inverse power method which is designed for the case where Q is a convex-convex ratio.

5.3 Convex function over convex function: Nonlinear inverse power method

We now consider the case where the functionals R and S in (5.1) are convex and p-homogeneous, for $p \geq 1$. Moreover, for the case $p > 1$, we need the additional assumption that S is continuous. Note that the ratio $\frac{R}{S}$ is in general non-convex and non-smooth. The nonlinear inverse power method (nonlinear IPM) was introduced in Hein and Bühler [2010] as a method to compute solutions of the associated nonlinear eigenproblem

$$0 \in \partial R(u) - \lambda \, \partial S(u) \tag{5.5}$$

(see Chapter 3). The method is a generalization of the standard inverse power method (see e.g. Golub and Van Loan [1996]), used to compute the smallest eigenvalue of a positive semi-definite matrix $A \in \mathbb{R}^{n\times n}$.

The main motivation for the nonlinear inverse power method is the observation that the linear system $Af^{k+1} = f^k$ solved in each step of the standard inverse power method can be rewritten as the optimization problem

$$f^{k+1} = \arg\min_u \left\{ \tfrac{1}{2}\langle u, Au\rangle - \langle u, f^k\rangle \right\}.$$

The direct generalization of the above optimization problem is given by

$$f^{k+1} = \arg\min_u \left\{ R(u) - \langle u, s(f^k)\rangle \right\}, \tag{5.6}$$

or equivalently, $0 \in r(f^{k+1}) - s(f^k)$, where $r(f) \in \partial R(f)$ and $s(f) \in \partial S(f)$. In the case of $p > 1$, one can use the direct generalization (5.6), as reported in Hein and Bühler [2010]. In Alg. 3 we give a slightly modified version of the algorithm originally reported in Hein and Bühler [2010]. The only difference is the additional factor λ^k in the inner problem. The equivalence of the two formulations follows from the following proposition.

Proposition 5.4. *Let $\Phi(f) := R(f) - \lambda \langle f, s\rangle$ and $\Psi(f) := R(f) - \langle f, s\rangle$ for a positively p-homogeneous function R for $p > 1$, $s \in \mathbb{R}^n$ and $\lambda > 0$. Then $\hat{f} \in \arg\min \Psi(f)$ if and only if $\alpha\hat{f} \in \arg\min \Phi(f)$, where $\alpha = \lambda^{\frac{1}{p-1}}$.*

Proof. Let $\hat{f} \in \arg\min \Psi(f)$, which implies that $\Psi(\hat{f}) \leq \Psi(f), \forall f \in \mathbb{R}^n$. Note that for $\alpha = \lambda^{\frac{1}{p-1}}$ it holds $\forall f \in \mathbb{R}^n$ that $\alpha^p \Psi(f) = \lambda^{\frac{p}{p-1}} \Psi(f) = \lambda^{\frac{p}{p-1}} R(f) - \lambda^{1+\frac{1}{p-1}} \langle f, s\rangle = \alpha^p R(f) - \lambda\alpha \langle f, s\rangle = \Phi(\alpha f)$, where we used the p-homogeneity of R. As $\alpha^p > 0$, this implies that $\Phi(\alpha\hat{f}) \leq \Phi(\alpha f), \forall f \in \mathbb{R}^n$, which again implies that $\alpha\hat{f} \in \arg\min \Phi(f)$. Analogously one shows the reverse direction. □

Thus the factor λ^k just leads to a rescaled solution of the inner problem. This change has been introduced in Alg. 3 as it will enable us to have a simpler convergence proof of the method.

Algorithm 3 Computing a nonlinear eigenvector for convex positively p-homogeneous functions R and S with $p > 1$

1: **Initialization:** $f^0 = \text{random}$, $\lambda^0 = Q(f^0)$
2: **repeat**
3: $\quad g^{k+1} = \arg\min_u \left\{ R(u) - \lambda^k \langle u, s(f^k)\rangle \right\}$ where $s(f^k) \in \partial S(f^k)$
4: $\quad f^{k+1} = g^{k+1} / S(g^{k+1})^{1/p}$
5: $\quad \lambda^{k+1} = Q(f^{k+1})$
6: **until** $\frac{|\lambda^{k+1} - \lambda^k|}{\lambda^k} < \epsilon$
7: **Output:** eigenvalue λ^{k+1} and eigenvector f^{k+1}.

As the direct generalization fails for $p = 1$, we present a modified method for this case in Alg. 4. The additional ball constraint $\|u\|_2 \leq 1$ needs to

be introduced as the objective of the inner problem would otherwise be unbounded from below. Note that the 2-norm is only chosen for algorithmic convenience, in principle, any norm can be chosen. Moreover, adding a ball constraint in Alg. 3 may potentially be harmful if it does not contain the optimal solution of the inner problem. However, as we will see in Lemma 5.5, it is possible if the ball contains a point satisfying the condition in Lemma 5.5. Furthermore, note that while the introduction of λ^k in Alg. 3 leads only to a rescaled solution of the inner problem, it is necessary in Alg. 4 to guarantee descent. Finally, note that the additional normalization in line 4 in Alg. 3 is not necessary to guarantee descent in the functional in each step. However it will be required later to show the convergence of the sequence f^k to the solution of a nonlinear eigenproblem, see Theorems 5.7 and 5.10.

Algorithm 4 Computing a nonlinear eigenvector for convex positively p-homogeneous functions R and S with $p = 1$

1: **Initialization:** $f^0 = \text{random with } \left\|f^0\right\| = 1,\ \lambda^0 = Q(f^0)$
2: **repeat**
3: $\quad f^{k+1} = \arg\min_{\|u\|_2 \leq 1} \left\{ R(u) - \lambda^k \left\langle u, s(f^k) \right\rangle \right\} \qquad \text{where } \ s(f^k) \in \partial S(f^k)$
4: $\quad \lambda^{k+1} = Q(f^{k+1})$
5: **until** $\frac{\left|\lambda^{k+1} - \lambda^k\right|}{\lambda^k} < \epsilon$
6: **Output:** eigenvalue λ^{k+1} and eigenvector f^{k+1}.

5.3.1 Monotonicity

Note that, in contrast to Dinkelbach's method, the inner problem does not use the function S but instead a linear lower bound. This makes the method applicable to the case where S is convex and enables us to prove convergence of the method. In the following, let $\Phi_{f^k}(u) := R(u) - \lambda^k \left\langle u, s(f^k) \right\rangle$ denote the objective of the convex inner problem in Alg. 3 and 4.

Lemma 5.5. *Let $g \in \mathbb{R}^n$ such that $\Phi_{f^k}(g) < R(f^k)\ (1-p)$ for Alg. 3 or $\Phi_{f^k}(g) < 0$ for Alg. 4. Then $Q(f^k) > Q(g)$.*

Proof. Note that for a positively p-homogeneous convex function one has

$$S(g) \geq S(f^k) + \left\langle s(f^k), g - f^k \right\rangle = \left\langle s(f^k), g \right\rangle + (1-p)\, S(f^k),$$

where we have used the convexity of S as well as the fact that $\left\langle s(f^k), f^k \right\rangle = p\, S(f^k)$ due to the p-homogeneity of S. Therefore we obtain

$$\Phi_{f^k}(g) = R(g) - \lambda^k \left\langle s(f^k), g \right\rangle \ \geq \ R(g) - \lambda^k S(g) + R(f^k)\ (1-p).$$

It follows that in both cases the condition in Lemma 5.5 implies that $R(g) - \lambda^k S(g) < 0$. Finally, we obtain $Q(g) = \frac{R(g)}{S(g)} < \lambda^k = Q(f^k)$. □

The following is an immediate corollary which shows that Alg. 3 and 4 create monotonically decreasing sequences.

Lemma 5.6. *The sequences f^k produced by Alg. 3 and 4 satisfy $Q(f^k) > Q(f^{k+1})$ for all $k \geq 0$ or the sequences terminate.*

Proof. It holds for $p \geq 1$ that

$$\Phi_{f^k}(f^k) = R(f^k) - \lambda^k \langle s(f^k), f^k \rangle = R(f^k) - \lambda^k \, p \, S(f^k) = R(f^k)\,(1-p),$$

where we have used that $\langle s(f^k), f^k \rangle = p\, S(f^k)$. Hence for both algorithms, the optimal value $\hat{f}$ of the inner problem satisfies $\Phi_{f^k}(\hat{f}) \leq R(f^k)\,(1-p)$. If equality holds, then f^k is a possible minimizer and the sequence terminates.

Otherwise, in the case $p = 1$ (Alg. 4), the optimal point $\hat{f} = f^{k+1}$ satisfies the condition from Lemma 5.5, which implies that $Q(f^{k+1}) < Q(f^k)$. In the case $p > 1$ (Alg. 3), one has $Q(g^{k+1}) < Q(f^k)$. The result then follows from $Q(f^{k+1}) = Q(g^{k+1})$ due to the p-homogeneity of R and S. □

Note that Lemmas 5.5 and 5.6 can be extended to arbitrary ratios of non-negative convex functions (not requiring p-homogeneity). However, since our definition of nonlinear eigenproblems requires the functions to be p-homogeneous, we restrict ourselves to the p-homogeneous case.

The importance of Lemma 5.5 arises from a practical consideration: Lemma 5.5 shows that descent in Q is not only guaranteed for the optimal solution of the inner problem, but for any vector u with $\Phi_{f^k}(u) < R(f^k)\,(1-p)$ in the case of Alg. 3 and $\Phi_{f^k}(u) < 0 = \Phi_{f^k}(f^k)$ for Alg. 4. This has two important practical implications. First, for the convergence of the IPM, it is sufficient to use a vector u satisfying the above conditions instead of the optimal solution of the inner problem. In particular, in an early stage where one is far away from the limit, it makes no sense to invest much effort to solve the inner problem accurately. Second, if the inner problem is solved by a descent method, a good initialization for the inner problem at step $k+1$ is given by f^k, as descent in Q is guaranteed after one step.

5.3.2 Relation to nonlinear eigenproblem

Finally, the following theorem shows the convergence of Algorithms 3 and 4 to a solution of a nonlinear eigenproblem as defined in (5.5).

Theorem 5.7 (Convergence of nonlinear IPM). *The sequences f^k of Alg. 3 and 4 have convergent subsequences that converge to an eigenvector f^* with eigenvalue $\lambda^* = \lim_{k\to\infty} Q(f^k) \in \big[0, Q(f^0)\big]$ in the sense that it solves the nonlinear eigenproblem (5.5). If S is continuously differentiable at f^*, then Q has a critical point at f^*.*

The nonlinear inverse power method was used in Hein and Bühler [2010] to derive methods for the Cheeger Cut problem as well as sparse PCA, see Chapters 7 and 10. In Section 5.4, we will present the RatioDCA, a generalization of the nonlinear IPM for ratios of non-negative d.c. function, which will allows us to solve an even larger class of problems. As in the case of the nonlinear IPM, one can guarantee the monotonicity of the sequence generated by the algorithm. Moreover, one can show the convergence of RatioDCA to a solution of the general form of nonlinear eigenproblems defined in Section 3. For this reason we will postpone the proof of Theorem 5.7 to Section 5.4 where we will state the more general statement for the RatioDCA, which will include the statement in Theorem 5.7 as special case.

5.4 The general case: RatioDCA

We now treat the general case of non-negative functions R and S. A problem of this type arises e.g. as the result of a tight relaxation of a constrained fractional set program, see Chapter 4. In this case, the fact that the functions R and S are the Lovász extensions of set functions $\widehat{R}, \widehat{S}$ implies that they are 1-homogeneous, see Prop. 2.13. Moreover, Prop. 2.21 implies that (5.1) can be written as ratio of differences of convex functions (d.c.), i.e. $R = R_1 - R_2$ with R_1, R_2 convex, and similarly for S. As the proof of Prop. 2.21 is constructive, the explicit form of this decomposition can be calculated. The considered problem can be written as

$$\min_{f \in \mathbb{R}^n} \frac{R(f)}{S(f)} \;=\; \min_{f \in \mathbb{R}^n} \frac{R_1(f) - R_2(f)}{S_1(f) - S_2(f)}. \tag{5.7}$$

The method RatioDCA has recently been proposed for minimizing a non-negative ratio of 1-homogeneous d.c. functions [Hein and Setzer, 2011]. We will show the connection to nonlinear eigenproblems of the form

$$0 \in \partial R_1(f) - \partial R_2(f) - \lambda(\partial S_1(f) - \partial S_2(f)). \tag{5.8}$$

Since the problems in Theorem 4.2 and 4.6 require optimization over the positive orthant, we also consider a variant of problem (5.7) where the optimization is done over the positive orthant. In this case we need to use a slight modification of the algorithm reported in Hein and Setzer [2011]. Both versions are given in Alg. 5. The difference lies in the inner problem in line 3, where we either have the constraint $u \in \mathbb{R}^n_+$ or $u \in \mathbb{R}^n$, depending on the constraint in our original optimization problem. One can easily see that if R and S are convex, the algorithm boils down to the nonlinear IPM from Section 5.3 for $p = 1$.

In Alg. 6 we present a modification of the RatioDCA for a ratio of a difference of convex p-homogeneous functions $R = R_1 - R_2$ and $S = S_1 - S_2$ for $p > 1$. Moreover, here we additionally assume S to be continuous. In

Algorithm 5 RatioDCA - Minimization of a non-negative ratio of 1-homogeneous d.c functions over $\mathbb{R}^n_+$ or $\mathbb{R}^n$

Initialization: $f^0 \in \mathbb{R}^n_+$, $\lambda^0 = Q(f^0)$
repeat
  $f^{k+1} = \underset{u \in \mathbb{R}^n_+ \ / \ u \in \mathbb{R}^n, \ \|u\|_2 \leq 1}{\arg\min} \left\{ R_1(u) - \langle u, r_2(f^k) \rangle - \lambda^k \Big(\langle u, s_1(f^k) \rangle - S_2(u) \Big) \right\}$
  where $r_2(f^k) \in \partial R_2(f^k)$, $s_1(f^k) \in \partial S_1(f^k)$
  $\lambda^{k+1} = Q(f^{k+1})$
until $\frac{|\lambda^{k+1} - \lambda^k|}{\lambda^k} < \epsilon$

the case where R and S are convex, Alg. 6 reduces to the nonlinear inverse power method in Alg. 3.

Algorithm 6 RatioDCA - Minimization of a non-negative ratio of p-homogeneous d.c functions over $\mathbb{R}^n_+$ or $\mathbb{R}^n$

Initialization: $f^0 \in \mathbb{R}^n_+$, $\lambda^0 = Q(f^0)$
repeat
  $g^{k+1} = \underset{u \in \mathbb{R}^n_+ \ / \ u \in \mathbb{R}^n}{\arg\min} \left\{ R_1(u) - \langle r_2(f^k), u \rangle - \lambda^k \Big(\langle s_1(f^k), u \rangle - S_2(u) \Big) \right\}$
  where $r_2(f^k) \in \partial R_2(f^k)$, $s_1(f^k) \in \partial S_1(f^k)$
  $f^{k+1} = g^{k+1} / S(g^{k+1})^{1/p}$.
  $\lambda^{k+1} = Q(f^{k+1})$
until $\frac{|\lambda^{k+1} - \lambda^k|}{\lambda^k} < \epsilon$

As for the case of the nonlinear IPM from Section 5.3, the difference between Alg. 5 and Alg. 6 lies in the norm constraint of the inner problem in Alg. 5 which is necessary as otherwise the problem would be unbounded from below. However, the choice of the norm plays no role in the proof and any norm can be chosen.

5.4.1 Monotonicity

We now prove the monotonicity of Alg. 5 and Alg. 6, generalizing the corresponding statements for Alg. 3 and Alg. 4. In the following, let

$$\Phi_{f^k}(u) := R_1(u) - \langle u, r_2(f^k) \rangle - \lambda^k \Big(\langle u, s_1(f^k) \rangle - S_2(u) \Big)$$

denote the objective of the inner problem.

Lemma 5.8. *Let $g \in \mathbb{R}^n$ such that $\Phi_{f^k}(g) < 0$ for Alg. 5 or $\Phi_{f^k}(g) < \Phi_{f^k}(f^k)$ for Alg. 6. Then $Q(f^k) > Q(g)$.*

Proof. Note that for all non-negative d.c. functions $R_1 - R_2$ and $S_1 - S_2$,

$$\begin{aligned}\Phi_{f^k}(f^k) &= R_1(f^k) - \langle r_2(f^k), f^k\rangle - \lambda^k \left(\langle s_1(f^k), f^k\rangle - S_2(f^k)\right)\\ &= R_1(f^k) - R_2(f^k) - \lambda^k \left(S_1(f^k) - S_2(f^k)\right)\\ &\quad + R_2(f^k) - \langle r_2(f^k), f^k\rangle + \lambda^k \left(S_1(f^k) - \langle s_1(f^k), f^k\rangle\right)\\ &= R_2(f^k) - \langle r_2(f^k), f^k\rangle + \lambda^k \left(S_1(f^k) - \langle s_1(f^k), f^k\rangle\right).\end{aligned}$$

Note that in the special case where R_1, R_2, S_1, S_2 are 1-homogeneous, one obtains $\Phi_{f^k}(f^k) = 0$ and thus the condition in Lemma 5.8 can be rewritten as $\Phi_{f^k}(g) < \Phi_{f^k}(f^k)$ in both cases (Alg. 5 and Alg. 6). Moreover, $\forall g \in \mathbb{R}^n$,

$$\begin{aligned}\Phi_{f^k}(g) &= R_1(g) - \langle r_2(f^k), g\rangle - \lambda^k \left(\langle s_1(f^k), g\rangle - S_2(g)\right)\\ &\geq R_1(g) - R_2(g) + R_2(f^k) - \langle r_2(f^k), f^k\rangle\\ &\quad - \lambda^k \left(S_1(g) - S_1(f^k) + \langle s_1(f^k), f^k\rangle - S_2(g)\right)\end{aligned}$$

where we used that for a convex function one has for all $f, g \in \mathbb{R}^n_+$,

$$S(g) \geq S(f) + \langle s(f), g - f\rangle \Leftrightarrow -\langle s(f), g\rangle \geq -S(g) + S(f) - \langle s(f), f\rangle .$$

Rearranging of the teams leads to

$$\begin{aligned}\Phi_{f^k}(g) &\geq R_1(g) - R_2(g) - \lambda^k\Big(S_1(g) - S_2(g)\Big)\\ &\quad + R_2(f^k) - \langle r_2(f^k), f^k\rangle + \lambda^k \left(S_1(f^k) - \langle s_1(f^k), f^k\rangle\right)\\ &= R_1(g) - R_2(g) - \lambda^k\Big(S_1(g) - S_2(g)\Big) + \Phi_{f^k}(f^k) .\end{aligned}$$

This implies that in both cases, the condition $\Phi_{f^k}(g) < \Phi_{f^k}(f^k)$ implies that $R_1(g) - R_2(g) - \lambda^k (S_1(g) - S_2(g)) < 0$. Finally, one obtains

$$Q(g) = \tfrac{R_1(g) - R_2(g)}{S_1(g) - S_2(g)} < \lambda^k = Q(f^k),$$

which completes the proof. □

One can now show that the sequences f^k are monotonically decreasing.

Proposition 5.9. *The sequences f^k generated by Alg. 5 and Alg. 6 satisfy $Q(f^k) > Q(f^{k+1})$ for all $k \geq 0$ or the sequences terminate.*

Proof. Clearly, the optimal point f^{k+1} of the inner optimization problem in Alg. 5 satisfies $\Phi_{f^k}(f^{k+1}) \leq \Phi_{f^k}(f^k) = 0$. If equality holds, then f^k is a possible minimizer and the sequence terminates. Otherwise the optimal point satisfies the conditions of Lemma 5.8 which implies $Q(f^{k+1}) < Q(f^k)$.

Analogously, in Alg. 6 one gets $\Phi_{f^k}(g^{k+1}) \leq \Phi_{f^k}(f^k)$. Again, the sequence either terminates or we have with Lemma 5.8, $Q(g^{k+1}) < Q(f^k)$. The result then follows from $Q(f^{k+1}) = Q(g^{k+1})$ due to the p-homogeneity of the functions R_1, R_2, S_1, S_2.

Note that the above argument is independent of the choice of the first constraint ($u \in \mathbb{R}^n$ or $u \in \mathbb{R}^n_+$) in line 3 of Alg. 5 and 6. □

Lemma 5.8 implies that as in the case of the nonlinear IPM, one does not need to solve the inner problem to full accuracy to guarantee descent in the functional Q, see the discussion after Lemma 5.6.

Note that the proof of Lemma 5.8 does not use the p-homogeneity, which implies that the statement of the Lemma is also valid for any non-negative differences of convex functions $R_1 - R_2$ and $S_1 - S_2$. Moreover, in the proof of Prop. 5.9, the p-homogeneity is only used to infer from the fact that $Q(h^{k+1}) \leq Q(f^k)$ that also $Q(f^{k+1}) \leq Q(f^k)$. For this reason, omitting the normalization step $f^{k+1} = g^{k+1}/S(g^{k+1})^{1/p}$ in Alg. 6 (and the additional variable g^{k+1}) would lead to a variant of RatioDCA where one can guarantee descent for any ratio of non-negative d.c. functions. However, note that we will use the normalization step as well as the fact that numerator and denominator of Q are p-homogeneous in the proof of Theorem 5.10 when we show the convergence to a solution of a nonlinear eigenproblem.

5.4.2 Relation to nonlinear eigenproblem

The following theorem shows the connection to nonlinear eigenproblems as defined in (5.8). It has previously been proven in Hein and Setzer [2011] for the case $p = 1$ as well as Hein and Bühler [2010] for the case where R and S are convex p-homogeneous. Moreover, it generalizes Theorem 5.7.

Theorem 5.10 (Convergence of RatioDCA). *The sequences f^k generated by Alg. 5 and 6 have convergent subsequences that converge to an eigenvector f^* with eigenvalue $\lambda^* = \lim_{k\to\infty} Q(f^k) \in \big[0, Q(f^0)\big]$ in the sense that it solves the nonlinear eigenproblem* (5.8)*. If S, R_2 and S_2 are continuously differentiable at f^*, then Q has a critical point at f^*.*

We first give the following Lemma which will be used in the proof.

Lemma 5.11. *The sequences f^k generated by Algorithms 5 and 6 are contained in a compact set.*

Proof. In the case of Alg. 5, we have $\|f^k\|_2 \leq 1$ for every k, which immediately gives the result. In the case of Alg. 6, we have for all k,

$$1 = S(f^k) = S\left(\frac{f^k}{\|f^k\|_2}\|f^k\|_2\right) = \|f^k\|_2^p\, S\left(\frac{f^k}{\|f^k\|_2}\right) \geq \|f^k\|_2^p \inf_{\|f\|_2=1} S(f),$$

where we used the p-homogeneity of S. As S is continuous, the minimum $m := \inf_{\|f\|_2=1} S(f)$ is attained for some f on the unit sphere. Moreover, by assumption we have $S(f) = 0$ if and only if $f = 0$. Hence, it must hold that $m > 0$ and one obtains

$$\left\|f^k\right\|_2 \leq \left(\tfrac{1}{m}\right)^{\frac{1}{p}},$$

which implies the result for Alg. 6. □

Proof of Theorem 5.10. By Prop. 5.9 in both cases the sequence $Q(f^k)$ is monotonically decreasing. By assumption R and S are non-negative. Thus Q is bounded below by zero, which implies convergence towards a limit

$$\lambda^* = \lim_{k\to\infty} Q(f^k) \ .$$

For both algorithms, the sequences f^k are contained in a compact set (see Lemma 5.11), which implies the existence of a subsequence f^{k_j} converging to some element f^*. As the sequence $Q(f^{k_j})$ is a subsequence of a convergent sequence, it has to converge towards the same limit λ^*.

To prove the convergence towards a solution of the nonlinear eigenproblem, we first show that the limit f^* is a global minimizer of the functional Φ_{f^*} in the inner problem. To do this, we need to make a case distinction between Alg. 5 and Alg. 6. In the case of Alg. 5, the objective of the inner optimization problem is non-positive at the optimal point, as we have shown before. Assume now that $\min_{\|f\|_2^2\leq 1} \Phi_{f^*}(f) < 0$. Then the vector

$$f^{**} = \operatorname*{arg\,min}_{\|f\|_2^2\leq 1} \Phi_{f^*}(f)$$

satisfies $\Phi_{f^*}(f^{**}) < \Phi_{f^*}(f^*) = 0$, and thus by Lemma 5.8 one has $Q(f^{**}) < Q(f^*)$, which is a contradiction to the fact that the sequence $Q(f^k)$ has converged to $\lambda^* = Q(f^*)$. Thus it must hold that $\min_{\|f\|_2^2\leq 1} \Phi_{f^*}(f) = 0$, i.e. the function Φ_{f^*} is non-negative in the unit ball. Using the 1-homogeneity of Φ_{f^*}, one can even conclude that the function Φ_{f^*} is non-negative everywhere, and thus $\min_f \Phi_{f^*}(f) = 0 = \Phi_{f^*}(f^*)$.

In the case of Alg. 6, one has $\min_f \Phi_{f^*}(f) \leq \Phi_{f^*}(f^*)$. Analogously to before, assume now that $\Phi_{f^*}(f^{**}) < \Phi_{f^*}(f^*)$ at the optimal point f^{**}. Then Lemma 5.8 implies that $Q(f^{**}) < Q(f^*)$, which again yields a contradiction to the fact that the sequence $Q(f^k)$ has converged to λ^*.

Thus for both algorithms we have established the fact that the limit f^* of the sequence f^{k_j} is a global minimizer of Φ_{f^*}. This implies

$$\begin{aligned} 0 \in \partial\Phi_{f^*}(f^*) &= \partial R_1(f^*) - r_2(f^*) - \lambda^*(s_1(f^*) - \partial S_2(f^*)) \\ &\subset \partial R_1(f^*) - \partial R_2(f^*) - \lambda^*(\partial S_1(f^*) - \partial S_2(f^*) \ , \end{aligned}$$

which shows that f^* is an eigenvector with eigenvalue λ^*. Since this argument was independent of the choice of the subsequence, every convergent subsequence yields an eigenvector with the same eigenvalue λ^*. Clearly it holds that $\lambda^* \leq Q(f^0)$. The last statement follows from Theorem 3.6. □

5.4.3 The RatioDCA-Prox

In Jost et al. [2013], a variation of the RatioDCA called RatioDCA-Prox was proposed. It replaces the norm constraint in Alg. 5 by any compact set containing a neighborhood of 0, i.e. a constraint of the form $G(f) \leq 1$, where G is non-negative, convex and p-homogeneous. Then an additional proximal term $-c^k \langle u, g(f^k) \rangle$ is added to the objective of the inner problem, where $c^k \in \mathbb{R}_+$ is a non-negative sequence of parameters, and $g(f^k) \in \partial G(f^k)$.

It is easy to see that for $G(f^k) = \|f\|_2^2$ and $c^k = 0$ one recovers the RatioDCA in Alg. 5. Moreover, it was shown in Jost et al. [2013] that also recent algorithms for NCut and Cheeger cut clustering [Bresson et al., 2012a,b] are recovered as special cases.

Note that the proof of Theorem 5.10 only establishes the existence of a convergent subsequence of f^k. In addition to convergence of the function values λ^k, one can give the following convergence result for the RatioDCA-Prox.

Proposition 5.12 (Jost et al. [2013]). *If G is strictly convex and for all k, $c^k \geq \gamma$ for some $\gamma > 0$, then any sequence f^k produced by RatioDCA-Prox fulfills $\left\|f^{k+1} - f^k\right\|_2 \to 0$.*

This shows that either the sequence of iterates converges to an element f^* or the set of accumulation points is a connected subset of $\{f \in \mathbb{R}^n \mid G(f) \leq 1\}$. Note that the above result does not apply for the case where $c^k = 0$, i.e. RatioDCA. However, in practice no clear difference in performance between the case $c^k = 0$ and the general case is observed [Jost et al., 2013].

5.4.4 Quality guarantee for RatioDCA

Note that in general, convergence of the RatioDCA to the global optimum cannot be guaranteed. However, we can provide a *quality guarantee* for the case when the RatioDCA is applied to a tight relaxation of a constrained fractional set program (CFSP). Recall from Chapter 4 that in this case the ratio is given as

$$Q_\gamma(f) = \frac{R_1^L(f) - R_2(f) + \gamma \left(T_1^L(f) - T_2(f)\right)}{S_1(f) - S_2^L(f)},$$

corresponding to a set function of the form

$$\widehat{Q}_\gamma(C) = \frac{\widehat{R}(C) + \gamma\widehat{T}(C)}{\widehat{S}(C)} = \frac{\widehat{R}_1(C) - \widehat{R}_2(C) + \gamma\big(\widehat{T}_1(C) - \widehat{T}_2(C)\big)}{\widehat{S}_1(C) - \widehat{S}_2(C)},$$

where the original objective of the CFSP was given as

$$\widehat{Q}(C) = \frac{\widehat{R}(C)}{\widehat{S}(C)} = \frac{\widehat{R}_1(C) - \widehat{R}_2(C)}{\widehat{S}_1(C) - \widehat{S}_2(C)}.$$

The functions R_2, T_2 and S_1 are extensions and R_1^L, T_1^L and S_2 are the Lovász extensions of the corresponding terms in $\widehat{Q}_\gamma$, see Theorem 4.7. The following theorem shows that in this case, RatioDCA either improves a given feasible set or stops after one iteration. Earlier versions of the following result for specific problems have been reported in Hein and Bühler [2010], Hein and Setzer [2011] and Rangapuram and Hein [2012].

Theorem 5.13 (Quality guarantee for RatioDCA). *Let A be a feasible set and $\gamma \in \mathbb{R}_+^K$ such that $\forall i,\ \gamma_i\,\theta_i > \widehat{Q}(A)\max_{C\subset V}\widehat{S}(C)$. Let f^* denote the result of RatioDCA initialized with $\mathbf{1}_A$, and let C_{f^*} denote the set found by optimal thresholding of f^*. Either RatioDCA terminates after one iteration, or it holds that $\widehat{Q}(C_{f^*}) < \widehat{Q}(A)$, and the set C_{f^*} is feasible.*

Proof. Proposition 5.9 implies that the RatioDCA either directly terminates or produces a strictly monotonically decreasing sequence. In the latter case, using the strict monotonicity and the fact that optimal thresholding does not increase the objective (Lemma 4.1 + 4.3), one obtains

$$\widehat{Q}(A) \;=\; \widehat{Q}_\gamma(A) \;=\; Q_\gamma(\mathbf{1}_A) \;>\; Q_\gamma(f^*) \;\geq\; Q_\gamma(\mathbf{1}_{C_{f^*}}) \;=\; \widehat{Q}_\gamma(C_{f^*}).$$

Assume now that C_{f^*} is infeasible. Then, one can derive analogously to (4.10) and (4.11) in the proof of Lemma 4.5 that

$$\widehat{Q}_\gamma(C_{f^*}) \;>\; \widehat{Q}(A) \;=\; \widehat{Q}_\gamma(A),$$

which contradicts the fact that $\widehat{Q}_\gamma(A) > \widehat{Q}_\gamma(C_{f^*})$. Thus, C_{f^*} has to be feasible, and it holds that $\widehat{Q}(A) > \widehat{Q}(C_{f^*})$. □

The above theorem implies that all constraints of the original constrained fractional set program are fulfilled by the set C_{f^*} returned by RatioDCA.

One might criticize about the above statement that it does not guarantee an improvement in every case. However, it is clear that this cannot be achieved since the set A may already correspond to a critical point $\mathbf{1}_A$ of the objective Q_γ. In practice however, we often observe a strong improvement when our method is initialized with the solutions given by competing other methods, as we will see in the experiments.

As discussed in Chapter 4, there exists several possibilities to construct a tight relaxation, depending on the choice of the functions R_2, T_2 and S_1 (see Theorem 4.4). For example, in Section 7.4 we will present two different tight relaxations of the normalized cut criterion. Now the question arises how the choice of the extensions affects the performance of the RatioDCA.

In Lemma 4.3 it was shown that the Lovász extension is maximal in the class of 1-homogeneous extensions. This implies that the subdifferential is maximal for the Lovász extension. As observed by Jost et al. [2013], this suggests that the Lovász extension should lead to better performance. This was experimentally confirmed in Jost et al. [2013], where it was shown on several graphs that the Lovász extension consistently leads to better results in terms of the obtained objective value.

A crucial part of the algorithms is the efficient solution of the inner problem, which will be discussed in the next chapter. In Part III we will then use the algorithms discussed in this chapter to derive methods for a wide range of applications in network analysis and dimensionality reduction.

Chapter 6

Fast first order methods for the convex inner problem in RatioDCA

In the previous chapter we derived a scheme to compute solutions of a class of non-convex problems involving a ratio of functions $R, S : \mathbb{R}^n \to \mathbb{R}$. The main idea was a decomposition into a sequence of convex problems. In this chapter, we show how this inner problem can be efficiently solved globally optimal. Assume that we apply the RatioDCA to a problem of the form

$$\min_{f \in \mathbb{R}^n_+} \frac{R_1(f) - R_2(f)}{S_1(f) - S_2(f)} := Q(f), \tag{6.1}$$

where $R_1 - R_2$ and $S_1 - S_2$ are non-negative differences of convex functions. The inner problem solved at each step of the RatioDCA is then given as

$$\operatorname*{arg\,min}_{u \in \mathbb{R}^n_+, \|u\|_2 \leq 1} \left\{ R_1(u) - \langle u, r_2(f^k) \rangle + \lambda^k \left(S_2(u) - \langle u, s_1(f^k) \rangle \right) \right\}, \tag{6.2}$$

where $r_2(f^k) \in \partial R_2(f^k)$ and $s_1(f^k) \in \partial S_1(f^k)$. Since R_1 and S_2 are convex and the constraints define a convex set, the above problem is a convex optimization problem and can be solved globally optimal [Bertsekas, 1999].

Of course what is the most efficient way to solve the inner problem depends on the nature of the terms in the given problem instance. In many cases there exists an elegant way to compute solutions of the inner problem efficiently. For instance, in the case of the local clustering problem and the maximum density subgraph problem in Chapters 8 and 9, the inner problem has an equivalent smooth dual problem which can be solved very efficiently using Nesterov's method [Nesterov, 1983, Beck and Teboulle, 2009], which we will briefly review in this section. In the case of the sparse PCA problem in Chapter 10, it even has a closed form solution.

However, it is clear that in general, the above problem may be difficult to solve. For this reason, we now give a brief overview about some first order methods for convex (non-smooth) problems. In contrast to second-order methods such as Newton's method, the advantage of first-order methods is that they do not require the computation of the Hessian, which makes them suitable for large scale problems. Moreover, our objective is in general non-smooth. Note that we give only a brief introduction into the various methods, for a more detailed discussion, see e.g. Bertsekas [1999], Nesterov [2004], Bertsekas [2010], Combettes and Pesquet [2011] and references therein.

We start by discussing some general results about first order methods. Then we review the basic gradient and subgradient method for unconstrained minimization. Afterwards, we go one step further and present some methods for the optimization over a convex set, including the fast projected gradient method by Nesterov [1983]. Next we consider proximal splitting methods, as well as their primal-dual variants, which can be used if a certain decomposable structure of the problem can be exploited. Finally we discuss a class of methods called bundle and bundle-level methods.

A particular difficult situation may arise if the RatioDCA is applied to a tight relaxation of a constrained fractional set program (see Chapter 4). In this case, R_1 and S_2 are the Lovász extensions of the corresponding set functions $\widehat{R}_1$ and $\widehat{S}_2$. The problem may arise that R_1 and S_2 are not known in a closed form which can be handled in an optimization algorithm easily. However, in this situation one can exploit the connection between the Lovász extensions and the corresponding set functions, as we will explain in Section 6.5. This enables us to solve the above optimization problem if the Lovász extensions and their subdifferentials are not known in closed form, but the original set functions can be computed efficiently.

6.1 General results for first order methods

Let us start with some general considerations. The convex inner problem we will consider in the following can be stated in a very general form as

$$\min_{x \in C} F(x), \tag{6.3}$$

where F is a convex lower semi-continuous function and C is some closed convex set. In the following we will focus our studies on first order iterative methods for convex problems. Given an initial starting value $x^0 \in C$, an *iterative method* computes a sequence $x^1, x^2, \dots$ approximating the optimal solution of the problem, in each step using an *oracle* to gain more information about the problem. The term *order* refers to the type of oracle used in each step of the iterative algorithm. A zero order method uses only function evaluations in each step, while a first order method also uses the gradient,

or subgradients. In contrast, a second or higher order method also utilizes the values of the higher derivatives [Nemirovsky and Yudin, 1983].

As our considered problems are in general non-differentiable, second order methods such as the Newton method are not applicable. Moreover, while it is known that these methods typically need fewer steps to converge, they require the computation of the Hessian and the solution of a linear system in each step and thus typically have a much higher iteration cost [Bertsekas, 1999], which makes them not suitable for large-scale problems. Note that there exist recent developments of Newton-type methods which compute approximations of the true Hessian and thus reduce the per iteration cost and memory requirement significantly, see e.g. Schmidt et al. [2011]. However, a discussion of these results would be beyond the scope of this thesis, thus we restrict ourselves to first order methods in the following.

To evaluate the performance of an algorithm one considers the *convergence rate* of the algorithm, see Bertsekas [1999]. The rate of convergence can be evaluated either in terms of the distance to the optimal solution x^*, or in terms of the difference to the optimal value $F(x^*)$. We say that the method converges to the optimal value in $O(f(t))$, if in each step $t \geq 0$ of the iterative algorithm, the current iterate x^t satisfies

$$\left|F(x^t) - F(x^*)\right| \leq C\, f(t),$$

for some function $f : \mathbb{R} \to \mathbb{R}_+$ with $\lim_{t\to\infty} f(t) = 0$ and a constant $C > 0$. In other words, we compare the decay of the distance to the optimal value to the decay of the function $f(t)$. Similarly, convergence in $O(f(t))$ to the optimal solution means that for all $t \geq 0$,

$$\left\|x^t - x^*\right\| \leq C\, f(t).$$

One particular important case is if $f(t) = \omega^t$ for some $\omega \in (0, 1)$. Then we say that the method has *linear* convergence. Note that in this case we have

$$\lim_{k\to\infty} \frac{f(t+1)}{f(t)} = \omega, \tag{6.4}$$

i.e. the bound $f(t)$ drops by a factor of ω in each iteration. Clearly, smaller values of ω imply faster convergence. If the limit in (6.4) is zero, we speak of *superlinear* convergence. On the other hand, if the limit in (6.4) is 1, we say that the method has *sublinear* convergence. This is for example given for the function $f(t) = \frac{1}{t^\alpha}$, for some $\alpha > 0$.

A slightly different way to analyze the convergence is by studying how many iterations are necessary to compute a solution within an ε radius around the optimal solution. For instance, if the function values converge to the optimal solution in $O(\frac{1}{t^2})$, then a solution with error at most ε can be computed in $O(\frac{1}{\sqrt{\varepsilon}})$ iterations. Note that the converse only holds if the statement is made for every $\varepsilon > 0$ and not only a fixed ε.

The performance of a particular algorithm of course depends on several factors: the dimension of the problem, the form of the set C as well as the smoothness properties of the function F. Regarding the latter, we will later consider the following three different classes of functions:

- F is in general non-differentiable. We further assume that F is L-Lipschitz continuous, i.e. $\exists L > 0$ such that for all $x, y \in \mathbb{R}^n$ one has $\|F(x) - F(y)\|_2 \leq L \ \|x - y\|_2$, and the minimizer x^* exists and satisfies $\|x_i - x^*\| \leq R$ for some $R > 0$. We denote this class by $\mathcal{F}^0_{L,R}(\mathbb{R}^n)$.
- F is continuously differentiable with L-Lipschitz continuous gradient, i.e. $\exists L > 0$ such that for all $x, y \in \mathbb{R}^n$ one has $\|\nabla F(x) - \nabla F(y)\|_2 \leq L \ \|x - y\|_2$. We denote this class by $\mathcal{F}^{1,1}_L(\mathbb{R}^n)$.
- F is continuously differentiable with L-Lipschitz continuous gradient, and strongly convex with parameter μ. We denote this class by $\mathcal{S}^{1,1}_{\mu,L}(\mathbb{R}^n)$.

Assume that one develops a method to optimize a function of one of the above classes, which uses only first order information, as well as general knowledge about the given function class (i.e. the values of L and μ). The question one may ask is now: is there a limit on the convergence rate one can guarantee? In other words, assume we have a method with known convergence rate, is there a way to judge whether this is already the best one can achieve or it is possible to improve it further? It turns out that this question has been answered for several classes of problems [Nemirovsky and Yudin, 1983, Nesterov, 2004]. We now summarize the results for the above function classes. These general results will later enable us to put the performance of an individual method into context and allow for a systematic evaluation of the methods discussed later.

First we consider the class of problems $\mathcal{F}^{1,1}_L(\mathbb{R}^n)$. We assume that the iterative process creates a sequence of points which can be written as

$$x^t \in x^0 + \text{Span}\{\nabla F(x^0), \ldots, \nabla F(x^{t-1})\}, \quad t \geq 1. \tag{6.5}$$

If one wants to give a guarantee of the convergence rate of a particular method for the given function class, all members of the class need to be considered, in other words, the convergence rate of the method is equal to the worst convergence rate among all members of the given class. This implies that a lower bound on the worst-case complexity can be obtained by showing that for every value of t there exist a function which is a member of the given class and after t steps has an error higher than a given lower bound. This is the main idea of the following result [Nesterov, 2004].

Theorem 6.1 (Nesterov [2004]). *For any t, $1 \leq t \leq \frac{1}{2}(n-1)$, and any x^0 in $\mathbb{R}^n$ there exists a function $F \in \mathcal{F}^{1,1}_L(\mathbb{R}^n)$ such that for any first order*

method satisfying the assumption in (6.5) *we have*

$$F(x^t) - F(x^*) \geq \frac{3L \left\|x^0 - x^*\right\|^2}{32(t+1)^2},$$
$$\left\|x^t - x^*\right\|^2 \geq \frac{1}{8} \left\|x^0 - x^*\right\|^2,$$

where x^* *is the minimum of* F.

The above result implies that, assuming that the maximum number of iterations is not too large compared to the dimension n, the best worst-case guarantee on the convergence rate of a first order method one can give is $O(\frac{1}{t^2})$ for the optimization of a function of $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$. Moreover, as stated by Nesterov [2004], the assumption in (6.5) can also be avoided by a more complicated argument.

Let us point out that the above result does *not* imply that one can never achieve a better convergence rate for a particular function $F \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$, or even a complete subclass of $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$. It only says that the best rate one can *guarantee uniformly* over all members of the function class is $O(\frac{1}{t^2})$, since there exists at least one member of the class where a better rate can not be achieved. For all practical purposes, the possibility exists that the explicit function used in the proof of the above statement is not "representable" of the whole class, and for all functions appearing in practice, one can achieve a better convergence rate. On the other hand, it might be that the bound is too loose, i.e. there exist even "worse" functions and thus the actual worst-case convergence rate is much higher. However, assuming that both are not the case, the above theorem yields a useful estimate of the convergence rate one should aim for when designing an algorithm for the above function class (in fact we will see later that the bound of $O(\frac{1}{t^2})$ is tight).

We now consider the case of strong convexity, i.e. $F \in \mathcal{S}_{\mu,L}^{1,1}(\mathbb{R}^n)$. As before, one can derive a lower bound by considering a family of "bad" functions which for every t and every first order method achieves an error after t steps which is higher than a given lower bound. The following theorem gives such a bound for the case where the considered vector space is infinite dimensional.

Theorem 6.2 (Nesterov [2004]). *For any* $x_0 \in \mathbb{R}^\infty$ *and any constants* $L > \mu > 0$ *there exists a function* $F \in \mathcal{S}_{\mu,L}^{\infty,1}(\mathbb{R}^\infty)$ *such that for any first order method satisfying the assumption in* (6.5) *we have*

$$\left\|x^t - x^*\right\|^2 \geq \omega^{2t} \left\|x^0 - x^*\right\|^2,$$
$$F(x^t) - F(x^*) \geq \frac{\mu}{2} \omega^{2t} \left\|x_0 - x^*\right\|^2,$$

where x^* *is the minimum of* F *and* $\omega = \frac{\sqrt{Q_F}-1}{\sqrt{Q_F}+1}$, *where* $Q_F = \frac{L}{\mu}$ *is the condition number of* F.

Since $\omega \in (0,1)$, the above result implies that the best worst-case convergence rate one can hope for in the case of strongly convex functions is linear convergence. Moreover, note that, as mentioned by Nesterov [2004], a similar statement can be made for the finite-dimensional case.

Let us now consider the non-smooth case, i.e. $F \in \mathcal{F}^0_{L,R}(\mathbb{R}^n)$. In this case, we assume that the sequence created by the iterative scheme can be written as

$$x^t \in x^0 + \text{Span}\{g(x^0), \ldots, g(x^{t-1})\}, \quad t \geq 1, \tag{6.6}$$

where for $i \geq 0$, $g(x^i) \in \partial F(x^i)$. One can now give the following lower bound.

Theorem 6.3 (Nesterov [2004]). *For any $0 \leq t \leq n-1$ there exists a function $F \in \mathcal{F}^0_{L,R}(\mathbb{R}^n)$ such that for any first order method satisfying* (6.6) *we have*

$$F(x^t) - F(x^*) \geq \frac{LR}{2(1+\sqrt{t+1})},$$

where x^* is the minimum of F.

Again the above theorem implies that the best guarantee on the convergence rate one can give for a particular first order method and a class of optimization problems of the above form is $O(\frac{1}{\sqrt{t}})$. We are now ready to study various types of first order methods for problems of the above type.

6.2 Basic first order methods for convex problems

We begin by reviewing some basic methods for convex minimization problems. Let us first consider the unconstrained case, i.e. we have $C = \mathbb{R}^n$ in (6.3). We start by discussing the basic gradient descent method for the case where the function F is differentiable. Then we present the subgradient method for non-differentiable functions F. Next, we show how the methods can be extended to the general constrained case. Finally, we present the fast projected gradient method by Nesterov [1983], which achieves a faster convergence rate. Most of the results from this section can be found in Bertsekas [1999, 2010].

6.2.1 Gradient method

The *gradient method* or *steepest descent method* is an iterative scheme where in each iteration a step is performed in direction of the negative gradient at the given point. At step $t+1$, the current iterate x^t is updated as

$$x^{t+1} = x^t - \alpha^t \nabla F(x^t),$$

where $a^t > 0$ is some step size. The motivation for this iterative scheme is as follows: For any direction $d \in \mathbb{R}^n$, a first order Taylor expansion at the point $x + \alpha d$ is given as

$$F(x + \alpha d) \approx F(x) + \alpha \langle d, \nabla F(x) \rangle .$$

Thus, for sufficiently small $\alpha > 0$, we have $F(x+\alpha d) < F(x)$ if $\langle d, \nabla F(x) \rangle < 0$. Among all vectors $d \in \mathbb{R}^n$ with $\|d\|_2^2 = 1$, the inner product $\langle d, \nabla F(x) \rangle$ is minimized for $d = -\frac{\nabla F(x)}{\|\nabla F(x)\|_2}$. Thus, the negative gradient gives the direction of steepest descent locally at the given point x.

There exist several possibilities to compute the step size in each step. Ideally, one would use an exact line search to compute the step size α^t which minimizes the objective F along the direction d. Since this is not practical, one approach is to perform a limited line search, i.e. at step t, given current iterate x^t and direction d^t, one computes for a fixed scalar $s > 0$,

$$\alpha^t = \operatorname*{arg\,min}_{\gamma \in [0,s]} F(x^t + \gamma d^t), \tag{6.7}$$

for instance using bisection. To avoid many expensive evaluations of the objective, another approach is to use a *backtracking line search*, i.e. start with some step size and then successively reduce it until a significant descent is guaranteed. An example is the so-called *Armijo rule*. First one chooses fixed scalars $s > 0$ and β, σ with $0 < \beta, \sigma < 1$. Then, at step t, given current iterate x^t and direction d^t, one sets the step size as $\alpha^t = \beta^m s$, where m is the first non-negative integer m for which

$$F(x^t + \beta^m s d^t) - F(x^t) \leq \sigma \beta^m s \left\langle \nabla F(x^t), d^t \right\rangle . \tag{6.8}$$

Other possibilities exist, see Bertsekas [1999]. Using the negative gradient as descent direction and choosing the step size according to (6.7) or (6.8), one obtains a non-increasing sequence of function values $F(x^t)$, which, assuming the gradient is Lipschitz continuous, converges in $O(\frac{1}{t})$ to the global optimal value $F(x^*)$ of F. Moreover, if we additionally assume that F is strongly convex, then the distance to the optimal value decays in $O(\omega^{2t})$, where $\omega = \frac{L-\mu}{L+\mu}$, i.e. we obtain linear convergence [Nesterov, 2004].

In view of the theoretical results from the last section, one observes that for the case $F \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$, the convergence rate is much higher than the corresponding lower bound (i.e. worse). Assuming that the lower bound is not too loose, this implies that the method is far from optimal for the given problem class. For the strongly convex case, the result in Theorem 6.1 implies that one cannot achieve superlinear convergence. In that sense, the convergence rate of $O(\omega^t)$ for the strongly convex case is unimprovable. However, taking into account the explicit value of ω, we observe that there is still room for improvement. This can be seen by noting that the term in Theorem 6.1 can be rewritten as $\frac{L-\mu}{L+\mu+\sqrt{L}\sqrt{\mu}}$, which for large values of L is significantly smaller than the value of ω.

6.2.2 Subgradient method

We now consider the case where the functional F is non-differentiable. The basic idea of the *subgradient method* is to perform the iterative scheme

$$x^{t+1} = x^t - \alpha^t s(x^t),$$

where $s(x^t) \in \partial F(f^t)$ is an element of the subdifferential of F at x^t, and α^t is a sequence of step sizes. Note that while the above iterative scheme has the same form as the gradient descent method where the gradient is replaced by an element of the subdifferential, an important difference to the differentiable case is that a step in direction of the subgradient is not guaranteed to be a descent direction. However, one can show [Bertsekas, 2010] that for sufficiently small step sizes α^t, each step reduces the distance of the current iterate to the set of optimal solutions of F, i.e. for any $t \geq 0$,

$$\left\|x^{t+1} - x^*\right\|_2 \; < \; \left\|x^t - x^*\right\|_2 .$$

The choice of the step sizes is crucial for the convergence of the algorithm. In contrast to the gradient method, usually no step size selection is performed and the step sizes are fixed initially. Several different choices exist, see e.g. Bertsekas [2010]. For instance, for a constant step size $\alpha > 0$,

$$\min_{0 \leq i \leq t} F(x^i) - F(x^*) \; \leq \; \frac{\alpha c^2}{2} + \frac{\left\|x^0 - x^*\right\|_2^2}{2t\alpha},$$

where c is an upper bound on the norm of the subgradients. From this one concludes that $\forall \varepsilon > 0$, using the constant step size $\alpha = \frac{\varepsilon}{c^2}$, one obtains an ε-optimal solution in $O(\frac{1}{\varepsilon^2})$ steps, or equivalently, the distance to an ε-optimal solution decays with rate $O(\frac{1}{\sqrt{t}})$. Note that this does not imply convergence to the optimum, since the value of ε depends on the choice of α. For a diminishing step size such that $\alpha_t \to 0$ and $\sum_{t=0}^{\infty} \alpha^t = \infty$ (for example $\alpha_t = \frac{1}{t}$), the algorithm converges to the optimal value, i.e.

$$\lim_{t \to \infty} F(x_i) - F(x^*) = 0.$$

The subgradient method is known to converge slowly and therefore typically is not applied in practice. See Bertsekas [2010] for further details. In the next section, we present algorithms for the constrained optimization of F.

6.2.3 Projected gradient and subgradient method

Let us now discuss how the above methods can be extended to constrained problems, i.e. now C is some closed convex set $C \subset \mathbb{R}^n$. In the following denote by $P_C(x)$ the projection on the set C. First we assume that $F \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$. The *projected gradient method* performs the iterative scheme

$$x^{t+1} = P_C\left(x^t - \alpha^t \nabla F(x^t)\right). \tag{6.9}$$

Similarly to the unconstrained case, several possibilities for step size selection exist, for instance exact line search or backtracking line search, see Bertsekas [1999]. In both cases the projected gradient method achieves the same convergence rate of $O(\frac{1}{t})$ as the gradient method in the case of functions in $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$, see e.g. Bertsekas [2010], Beck and Teboulle [2009].

We now consider the optimization of a (in general) non-differentiable convex function $F \in \mathcal{F}_{L,R}^0(\mathbb{R}^n)$ over a general convex set C. Similarly to the differentiable case one defines a constrained version of the subgradient method. The *projected subgradient method* performs the iterative scheme

$$x^{l+1} = P_C\big(x^l - \alpha^l s(x^l)\big),$$

where $s(x^t) \in \partial F(f^t)$. The convergence analysis is similar to the unconstrained case, and one can show that the best guaranteed convergence rate is $O(\frac{1}{\sqrt{t}})$ [Bertsekas, 2010]. Since both the projected subgradient method as well as projected gradient method have slow convergence, they are not very useful in practice. In the next section we consider a modification of the projected gradient method achieving a better convergence rate.

6.2.4 Accelerated gradient projection method

Nesterov [1983] introduced a first order method for the optimization of a smooth function over a convex set which leads to a significant improvement of the convergence rate compared to the projected gradient method discussed above. The *fast projected gradient method* first performs a projected gradient step and then computes the next iterate as a weighted average with the previous iterate. The algorithmic scheme can be found in Alg. 7. Note that in the original paper, Nesterov [1983] used backtracking line search to obtain the step size.

Algorithm 7 Nesterov's fast projected gradient [Nesterov, 1983]

1: **Input:** Lipschitz-constant L of ∇F, step size $0 < \alpha \leq \frac{1}{L}$
2: **Initialization:** $\theta^0 = 1$, $x^0, y^0 \in \mathbb{R}^n$
3: **for all** t=0, 1, 2, ... **do**
4: $\quad x^{t+1} = P_C\left(y^t - \alpha \nabla f(y^t)\right)$
5: $\quad \theta^{t+1} = \frac{1+\sqrt{1+4(\theta^t)^2}}{2}$,
6: $\quad y^{t+1} = x^{t+1} + \frac{\theta^t - 1}{\theta^{t+1}}\Big(x^{t+1} - x^t\Big)$.
7: **end for**

By the specific choice of update step, a convergence rate of the functional values of $O\left(\frac{1}{t^2}\right)$ is achieved [Nesterov, 1983], which is optimal with respect to the lower bound from Theorem 6.1. While this is still sublinear, it is a significant improvement compared to the convergence rate of $O\left(\frac{1}{t}\right)$ of the

projected gradient method. The method was extended to a special non-smooth case by Beck and Teboulle [2009], see next section.

We will later use Nesterov's method to solve the inner problems appearing in the balanced graph cut problem (Chapter 7) as well as the local clustering and community detection application (Chapters 8 and 9).

6.3 Proximal splitting methods

In this section we consider optimization problems of the form

$$\min_{f\in\mathbb{R}^n} \; F(x) = f(x) + g(x), \tag{6.10}$$

where the functions f, g, $\mathbb{R}^n \to \mathbb{R}$ are convex lower semi-continuous functions. Moreover, for the moment we assume that $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$. Problems of the above form have many applications in machine learning or signal processing. Usually the term $f(x)$ represents a smooth data term, and the functional $g(x)$ represents a regularization functional. Examples include the LASSO [Tibshirani, 1994] or total variation based denoising and deblurring problems [Beck and Teboulle, 2009]. Moreover, the inner problem (6.2) can be written in the above form if either R_1 or S_2 are in $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$. To see this, note that every constrained optimization problem over a convex set C can be written as

$$\min_{f\in C} \; F(x) \quad = \quad \min_{f\in\mathbb{R}^n} \; F(x) + I_C(x), \tag{6.11}$$

where for any set C, the function $I_C(x)$ is defined by $I_C(x) = 0$, if $x \in C$, and $I_C(x) = \infty$, else. This implies that (6.11) is a special case of (6.10).

Recently there has been a strong interest in *proximal splitting methods* for (6.10), which we will discuss in this section. See e.g. Beck and Teboulle [2009], Combettes and Pesquet [2011] for further details.

6.3.1 Proximal gradient method

Note that the problem in (6.10) is in general non-differentiable. Thus Theorem 6.3 tells us that the best we can hope for by applying a first order method without exploiting the structure of the problem is a convergence rate of $O(\frac{1}{\sqrt{t}})$. However, one can use the following trick to achieve faster convergence. Note that the projection on a convex set can be written as

$$P_C(y) \; = \; \arg\min_{x\in\mathbb{R}^n} \; \left\{ I_C(x) + \tfrac{1}{2} \left\| x - y \right\|_2^2 \right\}.$$

This forms the motivation to define the *proximity operator* as a generalization of the projection on a convex set as follows [Moreau, 1962]:

$$\mathrm{prox}_g(y) \; = \; \arg\min_{x\in\mathbb{R}^n} \; \left\{ g(x) + \tfrac{1}{2} \left\| x - y \right\|^2 \right\}.$$

The above operator has a unique solution [Combettes and Pesquet, 2011]. Moreover, one easily sees that at the optimal point x^* of (6.10) one has for any $\gamma > 0$ [Combettes and Wajs, 2005],

$$x^* = \text{prox}_{\gamma f}(x^* - \gamma \nabla g(x^*)).$$

This motivates a generalization of the projected gradient method from Section 6.2.3, by replacing in the iterative scheme in (6.9) the projection operator by the proximity operator. The result is the *proximal gradient method*, which performs the following iterative scheme, see Beck and Teboulle [2009], Combettes and Pesquet [2011],

$$\begin{aligned} x^{t+1} &= \text{prox}_{\alpha^t g}\left(x^t - \alpha^t \nabla f(x^t)\right) \\ &= \underset{x \in \mathbb{R}^n}{\arg\min} \left\{ \alpha^t g(x) + \tfrac{1}{2} \left\| x - (x^t - \alpha^t \nabla f(y^t)) \right\|_2^2 \right\}. \end{aligned}$$

Methods using the above scheme are often called *forward-backward* splitting algorithms. If $f = 0$, the scheme reduces to the *proximal point algorithm*

$$x^{t+1} = \text{prox}_{\alpha^t g}\,(x_n),$$

used to minimize a non-differentiable function [Martinet, 1970]. For the special case of $f(x) = \|Ax - b\|_2^2$ and $g(x) = \|x\|_1$, one obtains the *iterative shrinkage-thresholding algorithm (ISTA)*, see e.g. Daubechies et al. [2004], Combettes and Wajs [2005].

Of course the proximal gradient method is only useful if the proximity operator can be computed efficiently. It turns out that in many cases appearing in practice, the proximity operator has a closed form solution, see Combettes and Pesquet [2011]. For instance, in the case of the L_1 norm $g(x) = \|x\|_1$, it is given by $\left(\text{prox}_g(y)\right)_i = \text{sign}(y_i)\max\{0, |y_i| - 1\}$, for all $i = 1 \dots n$.

Similarly to the basic gradient method, there exist several possibilities to select the step size in each step, for instance choosing a constant step size $0 < \alpha \leq \frac{1}{L}$ or using a backtracking line search. It turns out that in both cases, the proximal gradient method achieves the same convergence rate of $O(\frac{1}{t})$ as the projected gradient method from the last section [Beck and Teboulle, 2009]. In view of the general convergence results from Section 6.1, the question arises whether the method can be further improved to the optimal rate of $O(\frac{1}{t^2})$, which will be answered in the next section.

6.3.2 Accelerated proximal gradient method

Beck and Teboulle [2009] introduced a method called FISTA which leads to a significant improvement of the convergence rate compared to the proximal gradient method discussed before. The *fast iterative shrinkage-thresholding algorithm* extends Nesterov's method to the non-smooth setting from (6.10),

by replacing the projection operator by a proximity operator. FISTA first performs a forward step with step size $\frac{1}{L}$ and then a backward step using the proximity operator. The next iterate is then computed as weighted average with the previous iterate, using the scheme introduced by Nesterov [1983].

Algorithm 8 FISTA [Beck and Teboulle, 2009]

1: **Input:** Lipschitz-constant L of ∇F, step size $0 < \alpha \leq \frac{1}{L}$
2: **Initialization:** $\theta^0 = 1$, $x^0, y^0 \in \mathbb{R}^n$
3: **for all** t=0, 1, 2, ... **do**
4: $\quad x^{t+1} = \text{prox}_{\alpha g}\left(y^t - \alpha \nabla f(y^t)\right)$
5: $\quad \theta^{t+1} = \frac{1+\sqrt{1+4(\theta^t)^2}}{2}$,
6: $\quad y^{t+1} = x^{t+1} + \frac{\theta^t - 1}{\theta^{t+1}}\Big(x^{t+1} - x^t\Big)$.
7: **end for**

The general scheme for FISTA is given in Alg. 8 for a constant step size. In Beck and Teboulle [2009], the authors use $\alpha = \frac{1}{L}$. Moreover, they also give a variant where the step size is determined via backtracking line search, as well as a monotone version which guarantees that the function values are non-increasing. In all cases, FISTA achieves the same rate of convergence of $O\left(\frac{1}{t^2}\right)$ as Nesterov's method, which is a substantial improvement compared to the convergence rate of $O\left(\frac{1}{t}\right)$ of the standard proximal gradient method.

Note that Nesterov independently developed a method for the case of composite functions (6.10) in a technical report [Nesterov, 2007], which achieves the same optimal convergence rate of $O(\frac{1}{t^2})$. However, note that [Nesterov, 2007] uses a more complicated scheme where by accumulating information from the previous iterates it computes a sequence of estimate functions that approximate the function F, whereas Alg. 8 is conceptually simpler as it only performs one proximal gradient step and uses the last two iterates in each step.

6.3.3 Douglas-Rachford splitting

Note that above we assumed that the function f is continuously differentiable. We now drop this restriction, i.e. both f and g in (6.10) are allowed to be non-smooth. In this case the problem can be solved via *Douglas-Rachford splitting*. This technique, which can be traced back to the work of Douglas and Rachford [1956], Lions and Mercier [1979] and later Eckstein and Bertsekas [1992], requires the solution of two proximal gradient steps in each iteration. As shown by Combettes and Pesquet [2007], the optimality condition of the optimal point $x^* \in \mathbb{R}^n$ of (6.10) can $\forall \gamma > 0$ be written as

$$\begin{aligned} x^* &= \text{prox}_{\gamma g}(y^*), \qquad \text{where } y^* \in \mathbb{R}^n \text{ such that} \\ y^* &= \text{rprox}_{\gamma f}\left(\text{rprox}_{\gamma g}(y^*)\right), \end{aligned}$$

and rprox is defined as $\text{rprox}_f(x) := 2\,\text{prox}_f(x) - x$. This forms the motivation for the general form of the Douglas-Rachford splitting algorithm given in Alg. 9 [Combettes and Pesquet, 2011].

Algorithm 9 Douglas-Rachford splitting

1: **Initialization:** $\varepsilon \in (0, 1)$, $\gamma > 0$, $y^0 \in \mathbb{R}^n$
2: **for all** t=0, 1, 2, ... **do**
3: $x^t = \text{prox}_{\gamma g}(y^t)$
4: $\lambda^t \in [\varepsilon, 2 - \varepsilon]$
5: $y^{t+1} = y^t + \lambda^t\big(\text{prox}_{\gamma f}(2x^t - y^t) - x^t\big)$.
6: **end for**

The scheme in Alg. 9 can be shown to converge to a solution of (6.10) [Combettes and Pesquet, 2011]. Douglas-Rachford splitting was used for several applications in image processing, see e.g. Combettes and Pesquet [2007]. Moreover, several other algorithms are recovered as special cases, for instance the *alternating direction method of multipliers* [Gabay, 1983], as shown by Eckstein and Bertsekas [1992], and the *alternating Split Bregman algorithm* [Goldstein and Osher, 2009], as shown by Setzer [2011]. See e.g. Combettes and Pesquet [2007], Combettes and Pesquet [2011] for further details on Douglas-Rachford splitting.

6.3.4 Primal-dual proximal splitting methods

Recently there as been a strong interest in *primal-dual splitting methods* for convex problems [Zhu and Chan, 2008, Esser et al., 2010, Chambolle and Pock, 2011]. The idea of primal-dual algorithms is to alternate between steps which minimize the primal objective and maximize the dual objective.

The advantage of these types of approaches is that they give a natural stopping criterion of the algorithm if the duality gap is below some non-negative value. The disadvantage is that one might spend too much time optimizing the dual objective with high accuracy, while the primal variable is already sufficiently good. We now consider a class of problems of the form

$$\min_{x \in \mathbb{R}^n} \; f(Ax) + g(x), \tag{6.12}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ are convex, lower semi-continuous functions and $A \in \mathbb{R}^{m \times n}$. We can express the term $f(Ax)$ with respect to its convex conjugate as $f(Ax) = \max_{y \in \mathbb{R}^m} \; \langle Ax, y \rangle - f^*(y)$. Thus we obtain the following saddle point formulation of the original primal problem,

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} \; \langle Ax, y \rangle - f^*(y) + g(x). \tag{6.13}$$

Similarly, one obtains the dual problem

$$\max_{y \in \mathbb{R}^m} \; -g^*(-A^T y) - f^*(y). \tag{6.14}$$

The goal is to compute a solution (x^*, y^*) of (6.13) satisfying

$$\begin{aligned} Ax^* &\in \partial f^*(y^*) \\ -A^T y^* &\in \partial g(x^*). \end{aligned}$$

The basic idea of primal-dual splitting methods is to alternate in the optimization between steps which minimize (6.13) with respect to x and maximize (6.13) with respect to y. Applying forward-backward splitting to the optimization problems with respect to x and y then leads to the proximal steps

$$\begin{aligned} y^{t+1} &= \operatorname{prox}_{\sigma f^*}\left(y^t + \sigma A x^t\right) \\ x^{t+1} &= \operatorname{prox}_{\tau g}\left(x^t - \tau A^T y^{t+1}\right), \end{aligned} \tag{6.15}$$

where σ, τ are some step sizes. To compute the proximal operator of f^*, one can make use of Moreau's identity, which states that [Rockafellar, 1970]

$$\operatorname{prox}_{\tau f}(x) + \tau \operatorname{prox}_{\frac{1}{\tau} f^*}\left(\frac{x}{\tau}\right) = x.$$

The scheme in (6.15) is the classical *Arrow-Hurwicz method* [Arrow et al., 1964], which can be shown to converge to a saddle point (x^*, y^*) with worst-case convergence rate $O(\frac{1}{\sqrt{t}})$ [Chambolle and Pock, 2011].

6.3.5 Accelerated primal-dual splitting

The primal-dual algorithm of Chambolle and Pock [2011] was recently proposed as a method for various variational problems in image analysis such as deconvolution, motion estimation and segmentation. The authors consider for the problem in (6.13) the partial primal-dual gap

$$\begin{aligned} \mathcal{G}_{B_1 \times B_2}(x, y) = &\max_{y' \in B_2} \left\langle Ax, y' \right\rangle - f^*(y') + g(x) \\ &- \min_{x' \in B_1} \left\langle Ax', y \right\rangle - f^*(y) + g(x'), \end{aligned}$$

where $B_1 \subset \mathbb{R}^n$ and $B_2 \subset \mathbb{R}^m$. They then study a variant of the Arrow-Hurwicz type scheme in (6.15) and show that the partial primal-dual gap decays in rate $O(\frac{1}{t})$ for general functions f and g.

Then further conditions are imposed on f and g. First they consider the case where either f or g^* have Lipschitz continuous gradient. Note that this is equivalent to f^* or g being strongly convex (see e.g. Theorems 4.2.1 and 4.2.2 in Chapter 10 of Hiriart-Urruty and Lemaréchal [1996]). Since the forward steps are done with respect to f and g^*, the results from Section 6.1 suggest that the optimal convergence rate is $O(\frac{1}{t^2})$. Indeed, Chambolle and Pock [2011] propose a modification of their algorithm where through a clever choice of sequences σ_t, τ_t and weighting scheme they can prove that

Algorithm 10 Primal-dual algorithm of Chambolle and Pock [2011]

Initialization: $\tau_0, \sigma_0 > 0$ with $\tau_0 \sigma_0 L^2 \leq 1$, $\overline{x}^0 = x^0 \in \mathbb{R}^n$, $y^0 \in \mathbb{R}^m$
for all t=0, 1, 2, ... **do**
  $y^{t+1} = \text{prox}_{\sigma_t f^*}\left(y^t + \sigma_t A \overline{x}^t\right)$
  $x^{t+1} = \text{prox}_{\tau_t g}\left(x^t - \tau_t A^T y^{t+1}\right)$
  $\theta_t = \frac{1}{\sqrt{1+2\gamma\tau_t}}$, $\tau_{t+1} = \theta_t \tau_t$, $\sigma_{t+1} = \frac{\sigma_t}{\theta_t}$.
  $\overline{x}^{t+1} = x^{t+1} + \theta_t\left(x^{t+1} - x^t\right)$.
end for

the iterates x^t of the derived method under the stated conditions converge with convergence rate $O(\frac{1}{t^2})$ to a solution x^*. The resulting algorithm is shown in Alg. 10.

Next, they consider the case when f and g^* have Lipschitz continuous gradient, or equivalently, f^* and g are strongly convex. Note that this implies that, seen as a function of x, the objective in (6.13) is a strongly convex function, and seen as a function of y, it is strongly concave. They then give a variant of their method which converges in $O(\omega^{2t})$ to the optimal solution, where $\omega = \frac{1+\theta}{2+\mu}$, for $\mu \leq \frac{2\sqrt{\mu\delta}}{M}$ and $\frac{1}{1+\mu} \leq \theta \leq 1$, where δ, γ are the strong convexity parameters of f^* and g and M is the norm of A. Setting $\theta = \frac{1}{1+\mu}$ and $\mu = \frac{2\sqrt{\mu\delta}}{M}$, one obtains $\omega = \frac{1}{1+\mu} = \frac{M}{M+2\sqrt{\mu\delta}}$. Thus, for a matrix A with small norm and functions with large parameter of strong convexity, the method becomes very fast.

The primal-dual algorithm in Alg. 10 was used by Hein and Setzer [2011], Hein et al. [2013] to solve the inner problem appearing in the RatioDCA for various balanced graph cut and hypergraph cut problems.

6.4 Bundle methods

We now consider the optimization of a in general non-differentiable objective F and give a brief overview of a class of algorithms referred to as bundle methods, and their more recent variants, which are called bundle-level methods. We will later use these methods in Section 6.5 to derive a general purpose method to solve the inner problem in RatioDCA. The main idea of these methods is to maintain a "bundle" of information in each step, typically consisting of subgradients computed in previous iterations, which are then used in each step to compute an improving approximation on the functional to be minimized. We now give a brief overview, for a more detailed discussion of the results from this section, see e.g. Nesterov [2004], Belloni [2005], Bertsekas [2010], Lan [2013].

6.4.1 Cutting plane method

The basic *cutting plane method* [Cheney and Goldstein, 1959, Kelley, 1960] uses the fact that due to the subgradient inequality (see Section 2), one has

$$F(y) \geq F(x) + \big\langle s(x), y - x \big\rangle,$$

for all $x, y \in \mathbb{R}^n$ and $s(x) \in \partial F(x)$. Hence, given a set of points $(x_i)_{i=1}^l$ and the corresponding subgradients $(s(x_i))_{i=1}^l$, one can construct a piecewise-linear approximation $\mathrm{cp}_t(y)$ of R (called the *cutting plane model*) as follows:

$$\mathrm{cp}_t(y) := \max_{i=1,\ldots,t} \Big\{ F(x_i) + \big\langle s(x_i), y - x_i \big\rangle \Big\}.$$

The cutting plane method now uses this idea by maintaining a sequence of points and corresponding subgradients (called the *bundle*) to compute a sequence of approximations $(\mathrm{cp}_t)_{t=1,\ldots}$, each time choosing g_{t+1} as the optimum of the model (cp_t) over a compact set. Hence the method needs to solve a linear program in each iteration. Note that by construction, we have $\mathrm{cp}_t(x) \leq \mathrm{cp}_{t+1}(x) \leq \cdots \leq F(x)$ for all $x \in \mathbb{R}^n$. The cutting plane method converges to an optimal solution, yet slowly [Bertsekas, 2010].

6.4.2 Bundle methods

A significant improvement was achieved through the introduction of *bundle methods* (Lemaréchal [1977], Kiwiel [1983, 1990]), which can be seen as stabilized version of the cutting plane method. These methods also maintain a piecewise-linear approximation $\mathrm{cp}_t(f)$ to the original function F. However, the next iterate x^{t+1} is computed differently. Usually a sequence of center points $\hat{x}^t$ is maintained, and in each iteration the next vector x^{t+1} is computed as [Bertsekas, 2010],

$$x^{t+1} = \operatorname*{arg\,min}_{y \in \mathbb{R}^n} \Big\{ \mathrm{cp}_t(y) + \tfrac{\mu_t}{2} \big\| y - \hat{x}^t \big\|_2^2 \Big\} = \mathrm{prox}_{\frac{1}{\mu}\mathrm{cp}_t}(\hat{x}^t).$$

This leads to a stabilization of the method as it will avoid drastic changes and make the next iterate closer to the current prox-center $\hat{x}^t$. The trade-off between minimizing the cutting-plane approximation cp and staying close to the current center point $\hat{x}^t$ is controlled by the parameter μ_t.

Since it is not guaranteed that the next iterate leads to a decrease in the functional, an important feature of bundle-methods is the distinction between *serious steps* and *null steps*: after the computation of x^{t+1}, a check is made whether a significant decrease in the objective is achieved. If that is the case, the new prox-center $\hat{x}^{t+1}$ is set to x^{t+1}, if not, it remains unchanged, i.e. $\hat{x}^{t+1} = \hat{x}^t$. Thus by construction the sequence of functional values evaluated at the prox-centers $F(\hat{x}^t)$ is monotonically decreasing.

In Alg. 11 we give a basic version of the bundle method, see Belloni [2005]. Several variants of this algorithm have been proposed in the literature. For instance, Helmberg and Rendl [1997] proposed a version of the bundle method adapted for solving semidefinite programs, and Oliveira et al. [2011] developed a bundle method for the case of an inexact oracle, i.e. dealing with noisy estimates of function values and subgradient.

Algorithm 11 Basic bundle method

Initialization: $\delta > 0$, $m \in (0,1)$, $x^0 = \hat{x}^0$.
for all t=0, 1, 2, ... **do**
  $x^{t+1} = \arg\min_{y \in \mathbb{R}^n} \left\{ \mathrm{cp}_t(y) + \frac{\mu_t}{2} \left\| y - \hat{x}^t \right\|_2^2 \right\}$
  $\delta_t := F(\hat{x}^t) - \left(\mathrm{cp}_t(x^{t+1}) + \frac{\mu_t}{2} \left\| x^{k+1} - \hat{x}^t \right\|_2^2 \right)$
  if $\delta_t < \delta$: STOP
  if $F(\hat{x}^t) - F(x^{t+1}) \geq m\delta_k$: $\hat{x}^{t+1} = x^{t+1}$ (Serious step)
  else: $\hat{x}^{t+1} = \hat{x}^t$ (Null step)
  $\mathrm{cp}_{t+1}(y) := \max \left\{ \mathrm{cp}_t(y), F(x^{t+1}) + \left\langle s(x^{t+1}), y - x^{t+1} \right\rangle \right\}$.
end for

6.4.3 Bundle-level methods

The *bundle-level method* was first proposed by Lemaréchal et al. [1995]. The main difference to previous work is that it introduced the idea of *level sets* into bundle methods. As in the cutting plane method, in each step t first the piecewise linear approximation cp_t is optimized, yielding a lower bound $\underline{F}^t$ on the global minimum of F. On the other hand, an upper bound $\overline{F}^t$ is given by the best objective found so far. The bundle-level method now computes a new level l_t as a convex combination of $\underline{F}^t$ and $\overline{F}^t$, i.e. $l_t = \lambda \overline{F}^t + (1-\lambda)\underline{F}^t$ for some $\lambda \in (0,1)$. The next iterate is then computed as

$$x^{t+1} = \arg\min_{y \in \mathbb{R}^n} \left\{ \left\| y - x^t \right\|_2^2 \mid \mathrm{cp}_t(y) \leq l_t \right\}.$$

Alg. 12 shows the basic bundle-level method, see e.g. Nesterov [2004]. Setting $\lambda = 0$ yields the cutting plane method. On the other hand, for $\lambda = 1$ typically x^{t+1} will be set to x^t and no progress is made.

The bundle-level method computes an ε-optimal solution in $O\left(\frac{1}{\varepsilon^2}\right)$ for general non-smooth problems, where the constants depend on the choice of λ [Lemaréchal et al., 1995, Nesterov, 2004]. In fact, one can derive the optimal value of the parameter λ achieving the best iteration complexity, which is given as $\lambda = \frac{1}{2+\sqrt{2}}$ [Nesterov, 2004]. In contrast to previous approaches, two problems need to be solved in each step: a linear program to compute the minimum of the cutting plane model, and a quadratic program to compute

Algorithm 12 Basic bundle-level method

Initialization: x^0, $\overline{F}^0 = F(x^0)$, $\mathrm{cp}_0(y) = F(x^0) + \langle s(x^0), y - x^0 \rangle$
for all t=0, 1, 2, ... **do**
  $\underline{F}^t = \min_{x \in B} \left\{ \mathrm{cp}_t(x) \right\}$
  $l_t = (1-\lambda)\underline{F}^t + \lambda \overline{F}^t \quad$ for some $\lambda \in (0,1)$
  $x^{t+1} = \underset{x \in \mathbb{R}^n}{\arg\min} \left\{ \|x - x^t\|_2^2 \mid \mathrm{cp}_t(x) \leq l_t \right\}$
  $\mathrm{cp}_{t+1}(y) := \max \left\{ \mathrm{cp}_t(y), F(x^{t+1}) + \langle s(x^{t+1}), y - x^{t+1} \rangle \right\}$.
  $\overline{F}^{t+1} = \min \left\{ \overline{F}^t, F(x^{t+1}) \right\}$
end for

the projection on the level set. Several authors proposed variants of the bundle method tailored towards very large-scale optimization problems, see e.g. Ben-Tal and Nemirovski [2005]. A variant of the bundle-level method replacing the solution of the two subproblems by approximate computations was considered by Richtárik [2012], who showed that the convergence rate only increases by a small factor depending on the level of approximation.

Recently, there has been a strong interest in *universally optimal* black box methods which automatically adjust to the smoothness properties of the problem. Lan [2013] considered the class of functions with Hölder continuous gradient, i.e. there exist $\rho \in (0,1)$ and $L > 0$ such that $\|\nabla F(x) - \nabla F(y)\| \leq L \|x - y\|^\rho$, $\forall x, y \in \mathbb{R}^n$. For $\rho = 1$ one obtains smooth problems with Lipschitz continuous gradient, $\rho = 0$ corresponds to the non-smooth case, and for $\rho \in (0,1)$ one obtains an "intermediate" level of smoothness referred to as *weakly smooth* [Lan, 2013, Nesterov, 2004]. Lan [2013] presented a black box bundle-level type algorithm which achieves the optimal iteration complexity for this class of functions, which is given as $O(\frac{1}{\varepsilon^\alpha})$, where $\alpha = \frac{2}{1+3\rho}$ [Nesterov, 2014]. Thus, it is optimal for smooth, non-smooth and weakly smooth problems. Later, Nesterov [2014] proposed a number of gradient based techniques performing a similar automated adjustment to the Hölder parameters, without requiring an input of the user.

For the general non-smooth case, bundle methods are particularly appealing, since they do not require any explicit knowledge about the objective in the implementation. In the next section, we use the general bundle-level method of (6.16) to derive a general-purpose method to solve the inner problem appearing in RatioDCA.

6.5 General-purpose method for inner problem

In the previous sections we have discussed a number of different methods for the solution of the inner problem. The idea was to give the reader a toolbox

of different algorithms which can be applied to solve the particular instance of the inner problem, depending on its structure and smoothness properties. For smooth problems, fast projected gradient methods such as the method by Nesterov [1983] are usually applicable and provide fast convergence in a few steps. Moreover, if the inner problem can be written as a sum of a smooth term and a non-smooth term with an efficiently computable proximal operator, proximal splitting methods, in particular the method of Chambolle and Pock [2011], work well in practice.

However, it is clear that such explicit knowledge of the structure of the inner problem may not be available. Moreover, a problem appearing in practice is if the problem (6.1) has been derived from a tight relaxation (see Chapter 4), in which case R_1 and S_2 are the Lovász extensions of set functions $\widehat{R}_1$ and $\widehat{S}_2$. In practice, often the Lovász extensions are difficult to compute or not known in closed form. Also the smoothness parameters are in general unknown. Thus, in this case it is often difficult to develop a good algorithm to solve the particular instance of the inner problem.

For this reason, in the following we will show how the bundle-level method from Alg. 12 can be used to solve the inner problem in the general non-smooth case. The method requires in each step the computation of a subgradient of the inner objective. We will now demonstrate that the subgradients can be obtained without having any explicit closed form of the inner objective, and instead using only function evaluations of the original set objective. Thus, the resulting algorithm is a black-box method which does not require any explicit knowledge of the structure of the inner problem.

6.5.1 Computation of subgradient of inner objective

In the following, let $\Phi(x) = R_1(x) - \left\langle x, r_2(f^k)\right\rangle + \lambda^k \left(S_2(x) - \left\langle x, s_1(f^k)\right\rangle\right)$ denote the objective of the inner problem. We now describe how to compute a subgradient of Φ at a given $x \in \mathbb{R}^n$. The main idea is the following: Using Lemma 2.20, an element r_1 of the subdifferential of R_1 can be computed explicitly as

$$(r_1(x))_i = \widehat{R}_1(C_i) - \widehat{R}(C_{i+1}),\ \forall i = 1, \ldots, n\ ,$$

and similarly for r_2, s_1, s_2. Thus, the explicit form of R_1 is not required to compute a subgradient of R_1 at x. A subgradient $\phi(x) \in \partial\Phi(x)$ is then given as

$$\begin{aligned}\phi(x) &= r_1(x) - r_2(f^k) + \lambda^k\big(s_2(x) - s_1(f^k)\big)\\ &= \Big(r_1(x) + \lambda^k s_2(x)\Big) - \Big(r_2(f^k) + \lambda^k s_1(f^k)\Big)\,.\end{aligned}$$

The above computation requires sorting of the vector f in each iteration, which can be done in $O(n\,log(n))$ [Cormen et al., 2001]. Note that here, at

each (outer) step k of the algorithm (RatioDCA or nonlinear IPM), the right part $r_2(f^k)+\lambda^k s_1(f^k)$ of the above subgradient is fixed, whereas the left part $r_1(x) + \lambda^k s_2(x)$ changes in each (inner) iteration of the algorithm used to solve the inner problem. Moreover, using the 1-homogeneity of Φ, by Lemma 2.8 the inner objective can be efficiently computed as $\Phi(x) = \langle x, \phi(x)\rangle$. Thus the subgradients as well as the objective can be evaluated without explicitly knowing the Lovász extension at any time, and only using function evaluations of the original set objective.

To solve the inner problem, we use the bundle-level method given in Alg. 12. As we will see below, it turns out that it is advantageous to replace the 2-norm constraint in the inner problem by an ∞-norm constraint (see Section 5), since this will lead to subproblems in the bundle method which are easier to handle and empirically lead to faster convergence. The inner problem at step k is then given as

$$\min_{\|x\|_\infty \leq 1} \langle \phi(x), x\rangle \,. \tag{6.16}$$

Alg. 12 requires the solution of two subproblems in each iteration: a linear program which updates the current cutting plane model (line 3), as well as a quadratic program to compute a projection on a level set (line 5). We will discuss the solution of both subproblems in the following.

6.5.2 Solution of the linear program

At each step of the algorithm, we have to compute the optimum of the current cutting plane model. Let K denote the current size of the bundle. Then, the problem can be explicitly written as

$$\begin{aligned} \min_x \max_{i=1\ldots K} &\left\{\Phi(x^i) + \left\langle \phi(x^i), x - x^i\right\rangle\right\} \\ \text{subject to} : &\left\|x\right\|_\infty \leq 1. \end{aligned} \tag{6.17}$$

Due to the 1-homogeneity of the inner objective, the cutting plane model simplifies to $\max_{i=1\ldots K} \left\langle \phi(x^i), x\right\rangle$. Introducing the matrix $A \in \mathbb{R}^{K\times n}$ containing in each row an element of the bundle (a previous subgradient $\phi(x^i)$), the above problem can be reformulated as

$$\begin{aligned} \min_{x,t}\ & t \\ \text{subject to} :\ & Ax - t\mathbf{1} \leq 0 \\ & x - \mathbf{1} \leq 0 \\ & -x - \mathbf{1} \leq 0. \end{aligned} \tag{6.18}$$

Thus we obtain a linear program which can be solved using standard LP solvers. In our implementation we used MOSEK, which applies an interior point method [Andersen and Andersen, 2000].

6.5.3 Solution of quadratic program

The second subproblem in the bundle-level method is a projection on the level set of the cutting plane model. Explicitly it is given as

$$\begin{aligned} \min_x \ & \tfrac{1}{2}\|x-z\|_2^2 \\ \text{subject to} : \ & \mathrm{cp}_t(x) \le l \\ & \|x\|_\infty \le 1, \end{aligned}$$

where l is the current level, and z is the current prox-center. As in the case of the LP, we use the matrix $A \in \mathbb{R}^{K\times n}$ of current bundle information to rewrite this as

$$\begin{aligned} \min_x \ & \tfrac{1}{2}\|x-z\|_2^2 \\ \text{subject to} : \ & Ax - l\mathbf{1} \le 0 \\ & -x - \mathbf{1} \le 0 \\ & x - \mathbf{1} \le 0. \end{aligned} \tag{6.19}$$

The problem (6.19) is an optimization problem with n variables and $2n+K$ constraints. The problem dimension can be significantly reduced by going over to a dual problem. To derive the dual, we proceed in two steps.

Lemma 6.4. *The problem in* (6.19) *is equivalent to the dual problem*

$$\begin{aligned} \min_{\alpha,\beta,\gamma} \ & \tfrac{1}{2}\|z - A^T\alpha + \beta - \gamma\|_2^2 - \tfrac{1}{2}\|z\|_2^2 + l\alpha^T\mathbf{1} + \beta^T\mathbf{1} + \gamma^T\mathbf{1} \\ \text{subject to} : \ & \alpha, \beta, \gamma \ge 0. \end{aligned} \tag{6.20}$$

Proof. The Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^K_+ \times \mathbb{R}^n_+ \times \mathbb{R}^n_+ \to \mathbb{R}$ is given as

$$L(x,\alpha,\beta,\gamma) = \tfrac{1}{2}\|x-z\|_2^2 + \alpha^T(Ax - l\mathbf{1}) + \beta^T(-x-\mathbf{1}) + \gamma^T(x-\mathbf{1}).$$

One obtains the optimality condition for x: $x - z + A^T\alpha - \beta + \gamma = 0 \Leftrightarrow x = z - A^T\alpha + \beta - \gamma$. Plugging the expression for x into the Lagrangian yields

$$\begin{aligned}
&\tfrac{1}{2}\left\|z - A^T\alpha + \beta - \gamma - z\right\|_2^2 + \alpha^T\left(A\left(z - A^T\alpha + \beta - \gamma\right) - l\mathbf{1}\right) \\
&- \beta^T\left(z - A^T\alpha + \beta - \gamma + \mathbf{1}\right) + \gamma^T\left(z - A^T\alpha + \beta - \gamma - \mathbf{1}\right) \\
=&\tfrac{1}{2}\left\|A^T\alpha\right\|_2^2 + \tfrac{1}{2}\left\|\beta\right\|_2^2 + \tfrac{1}{2}\left\|\gamma\right\|_2^2 - \beta^T A^T\alpha + \gamma^T A^T\alpha - \gamma^T\beta \\
&+ z^T A^T\alpha - \left\|A^T\alpha\right\|_2^2 + \beta^T A^T\alpha - \gamma^T A^T\alpha - l\alpha^T\mathbf{1} \\
&- z^T\beta + \beta^T A^T\alpha - \left\|\beta\right\|_2^2 + \gamma^T\beta - \beta^T\mathbf{1} \\
&+ z^T\gamma - \gamma^T A^T\alpha + \gamma^T\beta - \left\|\gamma\right\|_2^2 - \gamma^T 1 \\
=&-\tfrac{1}{2}\left\| - A^T\alpha\right\|_2^2 - \tfrac{1}{2}\left\|\beta\right\|_2^2 - \tfrac{1}{2}\left\| - \gamma\right\|_2^2 + \beta^T A^T\alpha - \gamma^T A^T\alpha + \gamma^T\beta \\
&+ z^T A^T\alpha - z^T\beta + z^T\gamma - l\alpha^T\mathbf{1} - \beta^T\mathbf{1} - \gamma^T\mathbf{1} \\
=&-\tfrac{1}{2}\left\| - A^T\alpha + \beta - \gamma\right\|_2^2 + z^T A^T\alpha - z^T\beta + z^T\gamma - l\alpha^T\mathbf{1} - \beta^T\mathbf{1} - \gamma^T\mathbf{1} \\
=&-\tfrac{1}{2}\left\|z - A^T\alpha + \beta - \gamma\right\|_2^2 + \tfrac{1}{2}\left\|z\right\|_2^2 - l\alpha^T\mathbf{1} - \beta^T\mathbf{1} - \gamma^T\mathbf{1}.
\end{aligned}$$

Thus we obtain the dual problem in (6.20). □

We now have a problem with $2n + K$ variables and $2n + K$ inequality constraints. It turns out that we can simplify the above problem as follows.

Lemma 6.5. *The problem in* (6.20) *is equivalent to the problem*

$$\begin{aligned}
&\min_{\alpha} \tfrac{1}{2}\left\|v\right\|_2^2 - \tfrac{1}{2}\left\|P_{\mathbb{R}^n_+}(-\mathbf{1} - v)\right\|_2^2 - \tfrac{1}{2}\left\|P_{\mathbb{R}^n_+}(v - \mathbf{1})\right\|_2^2 - \tfrac{1}{2}\left\|z\right\|_2^2 + l\alpha^T\mathbf{1} \\
&\text{subject to} : \alpha \geq 0 \\
&\qquad\quad v = z - A^T\alpha.
\end{aligned} \tag{6.21}$$

Proof. We first perform a variable substitution $\delta = \beta - \gamma \Leftrightarrow \gamma = \beta - \delta$, where $\delta \in \mathbb{R}^n$ and $\delta \leq \beta$. Note that δ can now be negative. The problem (6.20) then becomes

$$\begin{aligned}
&\min_{\alpha,\beta,\delta} \tfrac{1}{2}\left\|z - A^T\alpha + \delta\right\|_2^2 - \tfrac{1}{2}\left\|z\right\|_2^2 + l\alpha^T\mathbf{1} + (2\beta - \delta)^T\mathbf{1} \\
&\text{subject to} : \alpha, \beta \geq 0, \beta \geq \delta.
\end{aligned} \tag{6.22}$$

We observe that the objective is linear in β and hence the optimum with respect to β is achieved at the boundary, i.e. $\beta = \max\{0, \delta\} =: P_{R^n_+}(\delta)$ and hence $2\beta - \delta = |\delta|$. We obtain the simplified problem

$$\begin{aligned}
&\min_{\alpha,\delta} \tfrac{1}{2}\left\|z - A^T\alpha + \delta\right\|_2^2 - \tfrac{1}{2}\left\|z\right\|_2^2 + l\alpha^T\mathbf{1} + \left\|\delta\right\|_1 \\
&\text{subject to} : \alpha \geq 0.
\end{aligned} \tag{6.23}$$

Next we observe that the problem is separable in the $\delta_i, i = 1 \dots n$. Let $v := z - A^T\alpha$. We have to solve for each δ_i the problem

$$\delta_i^* := \arg\min_{\delta_i} \tfrac{1}{2}(v_i + \delta_i)^2 + |\delta_i|.$$

At the optimal value one has $0 \in \partial\left(\frac{1}{2}(v_i + \delta_i)^2 + |\delta_i|\right)$ and hence $0 \in v_i + \delta_i^* + \operatorname{sign}(\delta_i^*)$. Performing a case distinction on v, one obtains

$$\begin{aligned}\delta_i^* &= \begin{cases} -1 - v_i, & \text{if } v_i < -1 \\ 0, & \text{if } -1 \le v_i \le 1 \\ 1 - v_i, & \text{if } 1 < v_i \end{cases} \\ &= P_{\mathbb{R}^n_+}(-1 - v_i) - P_{\mathbb{R}^n_+}(v_i - 1).\end{aligned}$$

We can now eliminate the variable δ in (6.23) and obtain the problem

$$\begin{aligned}\min_{\alpha}\ & \tfrac{1}{2}\left\|v + P_{\mathbb{R}^n_+}(-\mathbf{1} - v) - P_{\mathbb{R}^n_+}(v - \mathbf{1})\right\|_2^2 - \tfrac{1}{2}\|z\|_2^2 + l\alpha^T\mathbf{1} \\ &+ \left\langle P_{\mathbb{R}^n_+}(-\mathbf{1} - v) + P_{\mathbb{R}^n_+}(v - \mathbf{1}), \mathbf{1}\right\rangle \\ \text{subject to}:\ & \alpha \ge 0 \\ & v = z - A^T\alpha.\end{aligned} \tag{6.24}$$

We can further rewrite the objective as follows

$$\begin{aligned}&\tfrac{1}{2}\|v\|_2^2 + \tfrac{1}{2}\|P_{\mathbb{R}^n_+}(-\mathbf{1} - v)\|_2^2 + \tfrac{1}{2}\|P_{\mathbb{R}^n_+}(v - \mathbf{1})\|_2^2 \\ &+ \left\langle P_{\mathbb{R}^n_+}(-\mathbf{1} - v), v\right\rangle - \left\langle P_{\mathbb{R}^n_+}(v - \mathbf{1}), v\right\rangle - \left\langle P_{\mathbb{R}^n_+}(-\mathbf{1} - v), P_{\mathbb{R}^n_+}(v - \mathbf{1})\right\rangle \\ &+ \left\langle P_{\mathbb{R}^n_+}(-\mathbf{1} - v) + P_{\mathbb{R}^n_+}(v - \mathbf{1}), \mathbf{1}\right\rangle - \tfrac{1}{2}\|z\|_2^2 + l\alpha^T\mathbf{1} \\ =\ &\tfrac{1}{2}\|v\|_2^2 + \tfrac{1}{2}\|P_{\mathbb{R}^n_+}(-\mathbf{1} - v)\|_2^2 + \tfrac{1}{2}\|P_{\mathbb{R}^n_+}(v - \mathbf{1})\|_2^2 \\ &+ \left\langle P_{\mathbb{R}^n_+}(-\mathbf{1} - v), v + \mathbf{1}\right\rangle - \left\langle P_{\mathbb{R}^n_+}(v - \mathbf{1}), v - \mathbf{1}\right\rangle - \tfrac{1}{2}\|z\|_2^2 + l\alpha^T\mathbf{1} \\ =\ &\tfrac{1}{2}\|v\|_2^2 - \tfrac{1}{2}\|P_{\mathbb{R}^n_+}(-\mathbf{1} - v)\|_2^2 - \tfrac{1}{2}\|P_{\mathbb{R}^n_+}(v - \mathbf{1})\|_2^2 - \tfrac{1}{2}\|z\|_2^2 + l\alpha^T\mathbf{1}.\end{aligned}$$

Thus we obtain the problem in (6.21). □

We now have obtained a problem with K variables and K inequality constraints. Note that typically the bundle size K is much smaller than the dimension n. To solve the problem, we make use of the fact that the objective is differentiable, as we will show below. We can then solve it very efficiently using Nesterov's method, see Section 6.2.4. First we need the following lemma.

Lemma 6.6. *The function $F(v) := \frac{1}{2}\|P_{\mathbb{R}^n_+}(v)\|_2^2$ is differentiable for all $v \in \mathbb{R}^n$ and it holds that $\nabla F(v) = P_{\mathbb{R}^n_+}(v)$.*

Proof. Note that in order to prove that F is differentiable, it is sufficient to prove that all the partial derivatives exist and are continuous for all $v \in \mathbb{R}^n$ (see e.g. Theorem 1.9.5 in Hubbard and Hubbard [1998]). Thus, the problem reduces to the differentiability of $\frac{1}{2}\max\{0, v_k\}^2$, for all $k = 1, \ldots, n$. Clearly, the objective is differentiable if $v_k < 0$ and $v_k > 0$ and the derivative is given by 0 and v_k, respectively. For the case $v_k = 0$, we obtain

$$\lim_{h \searrow 0} \frac{\max\{0, v_k + h\}^2 - \max\{0, v_k\}^2}{h} = \lim_{h \searrow 0} \frac{h^2 - 0^2}{h} = 0.$$

On the other hand,

$$\lim_{h \nearrow 0} \frac{\max\{0, v_k + h\}^2 - \max\{0, v_k\}^2}{h} = \lim_{h \nearrow 0} \frac{0^2 - 0^2}{h} = 0.$$

Thus both one-sided limits agree and the partial derivative exists at 0. In total, the k-th partial derivative is given by $\max\{0, v_k\}$, which is a continuous function. □

We can now use the above lemma to show the differentiability of the objective in (6.21). In the following, $P_{B_\infty(1)}(v)$ denotes the projection on the L_∞ unit ball, given as $B_\infty(1) = \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 1\}$.

Lemma 6.7. *The objective $\Psi(\alpha)$ in (6.21) is differentiable for all $\alpha \in \mathbb{R}^K_+$ and it holds that*

$$\nabla_\alpha \Psi(\alpha) = -A P_{B_\infty(1)}(v) + l\mathbf{1}$$

where $v = z - A^T\alpha$. Moreover, an upper bound on the Lipschitz constant of the gradient is given by $\|A\|_F^2$.

Proof. We first interpret the inner objective as a function of the variable v and compute the gradient of Ψ with respect to v. Using Lemma 6.6 and the chain rule, we obtain

$$\nabla \Psi_v(v) = v + P_{\mathbb{R}^n_+}(-\mathbf{1} - v) - P_{\mathbb{R}^n_+}(v - \mathbf{1}) = P_{B_\infty(1)}(v). \tag{6.25}$$

We now compute the gradient of Ψ with respect to α. Again using the chain rule as well as (6.25) and the fact that $v = z - A^T\alpha$, one obtains

$$\nabla \Psi_\alpha(\alpha) = -A P_{B_\infty(1)}(z - A^T\alpha) + l\mathbf{1}.$$

Regarding the Lipschitz constant , one obtains for $\alpha_1, \alpha_2 \in \mathbb{R}^K_+$,

$$\begin{aligned}
&\|\nabla \Psi(\alpha_1) - \nabla \Psi(\alpha_2)\|_2^2 \\
&= \left\| -A P_{B_\infty(1)}(z - A^T\alpha_1) + l\mathbf{1} + A P_{B_\infty(1)}(z - A^T\alpha_2) - l\mathbf{1} \right\|_2^2 \\
&= \left\| -A \left(P_{B_\infty(1)}(z - A^T\alpha_1) - P_{B_\infty(1)}(z - A^T\alpha_2) \right) \right\|_2^2 \\
&\leq \|A\|_F^2 \left\| P_{B_\infty(1)}(z - A^T\alpha_1) - P_{B_\infty(1)}(z - A^T\alpha_2) \right\|_2^2 \\
&\leq \|A\|_F^2 \left\| (z - A^T\alpha_1) - (z - A^T\alpha_2) \right\|_2^2 \\
&\leq \|A\|_F^2 \left\| A^T \right\|_F^2 \|\alpha_1 - \alpha_2\|_2^2 = \|A\|_F^4 \|\alpha_1 - \alpha_2\|_2^2,
\end{aligned}$$

and thus an upper bound on the Lipschitz constant is given by $\|A\|_F^2$. □

Using Lemma 6.7, the quadratic problem can now be solved efficiently using Nesterov's method, see Alg. 6.5.3.

Nesterov's method for the QP appearing in the bundle-level method

Input: Lipschitz constant L of $\nabla\Psi$,
Initialization: $\theta_1 = 1$, $\alpha^1, \beta^1 \in \mathbb{R}_+^K$,
repeat
$v^t = z - A^T\alpha^t$
$\nabla = -A\left(P_{B_\infty(1)}(v^t)\right) + l\mathbf{1}$
$\beta^{t+1} = P_{\mathbb{R}^n_+}(\alpha^t - \frac{1}{L}\nabla)$
$\theta_{t+1} = \frac{1+\sqrt{1+4\theta_t^2}}{2}$,
$\alpha^{t+1} = \beta^{t+1} + \frac{\theta_t - 1}{\theta_{t+1}}\Big(\beta^{t+1} - \beta^t\Big)$.
until duality gap $< \epsilon$

Finally, note that since we are adding a new element to the matrix A in each step of the bundle method, the computational time as well as memory requirement to solve the two subproblems increases in each iteration, which becomes a problem when working with large bundle sizes. However, in practice it is not necessary to maintain the full cutting plane model in each iteration. Instead one uses a slightly less accurate model which only uses the last B subgradients, and keeps track of the best lower bound found so far in each step. In our experiments, a value of $B = 50$ empirically lead to a good trade-off between iteration cost and number of iterations.

Equipped with the results from Chapter 4 as well as the algorithms from the last two chapters, we are now able to derive methods for a large class of applications in network analysis and dimensionality reduction.

Part III

Applications in network analysis and dimensionality reduction

Chapter 7

Balanced graph partitioning and 1-Spectral Clustering

The problem of balanced graph partitioning has a wide range of applications from circuit design to image segmentation [Hagen and Kahng, 1991, Shi and Malik, 2000]. Applied to a graph representing pairwise similarities between data points, it quite naturally leads to a technique for data clustering. The most popular method for graph-based data clustering in the machine learning community is spectral clustering [Shi and Malik, 2000, Meila and Shi, 2001, Ng et al., 2001, Ding et al., 2001, von Luxburg, 2007]. In spectral clustering, the NP-hard graph partitioning problem is relaxed to a standard linear eigenproblem involving the so-called graph Laplacian.

Recently, we proposed the method p-spectral clustering [Bühler, 2009, Bühler and Hein, 2009a], where we showed that better bounds on the obtained cut values can be obtained by a relaxation to a nonlinear eigenproblem involving the graph p-Laplacian, and letting $p \to 1$. The logical next step was then to consider the case $p = 1$ directly and study the eigenproblem associated to the graph 1-Laplacian, which then leads to the method 1-spectral clustering [Hein and Bühler, 2010, Hein and Setzer, 2011]. One can show that in this case one obtains a tight relaxation of the corresponding balanced cut criterion. The resulting algorithm outperforms standard spectral clustering in terms of the obtained cut objective by a large margin.

In this chapter we give an overview about the above methods for balanced graph cuts. We begin by discussing different graph cut criteria. Then in Section 7.2 we review the standard spectral relaxation of the normalized cut problem. Next we review the p-spectral relaxation and show that it leads to better bounds on the obtained cut values, which is done in Section 7.3. Then, in Section 7.4, we discuss 1-spectral clustering, which can now be derived directly using the tight relaxation framework discussed in Chapter 4. Finally we experimentally demonstrate the superiority of 1-spectral clustering to the standard relaxation and other approaches.

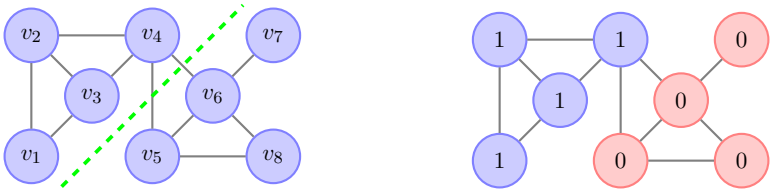

Figure 7.1: *Left:* Clustering as graph partitioning. *Right:* Equivalent graph labeling to the partition on the left.

7.1 Clustering via graph cuts

Given a set of points in a feature space, in clustering one is interested in finding groups of points in the data which are similar to each other (referred to as clusters). The idea of graph-based clustering is to represent the data as weighted, undirected graph $G(V, E, W)$ with sets of vertices V and edges E, where the vertices in V correspond to points in the feature space and the entry w_{ij} of the weight matrix $W \in R^{n\times n}$ for $n = |V|$ represents the similarity between the points i and j. Moreover, we will denote the degrees of the vertices in V by $d_i = \sum_{j\in V} w_{ij}$, for every $i \in V$. The matrix $D \in \mathbb{R}^{n\times n}$ is then the diagonal matrix having the degrees on the diagonal. Moreover, we denote by $\mathrm{vol}(C) = \sum_{i\in C}$ the volume of a given set $C \subset V$.

The clustering problem can then be formulated as graph partitioning problem: the task is to find a partition of the vertex set V into subsets $C_1, \dots C_k$ such that a given criterion is optimized. Ideally one has for each cluster a high within-cluster similarity and high dissimilarity between clusters. Partitioning is done by removing edges, referred to as a cut. Equivalently, this can also be seen as a graph labeling problem, see Figure 7.1. In the following we discuss some common optimization criteria for the graph partitioning problem. Note that throughout this chapter, we will assume that the graph is connected.

7.1.1 Unbalanced graph cuts

In unbalanced graph partitioning, the goal is to find a partition such that the connectivity between different clusters is small. This is achieved by minimizing the *cut* between the two clusters, given as

$$\mathrm{cut}(C, \overline{C}) = \sum_{i\in C, j\in\overline{C}} w_{ij}.$$

This problem can be efficiently solved in polynomial time, see e.g. Cormen et al. [2001]. The usual approach is to use its connection to the problem of finding the *maximum flow* in a network. A flow network is a directed

graph with a designated source node s and sink node t and non-negative capacities on each edge. The maximum flow problem now asks for the maximum amount of "material" which can be "transported" from s to t while taking into account the constraints imposed by the capacity of each edge. The Max-Flow-Min-Cut Theorem [Ford and Fulkerson, 1956] states that the value of the maximum flow on the s-t-graph is equal to the minimum s-t-cut on the graph. Given our graph G, the minimum cut problem can now be reduced to $|V| - 1$ s-t-cut problems by fixing a vertex s and considering all possible choices for the vertex t.

The classical method by Ford and Fulkerson [1956] finds the maximum flow by the following iterative procedure: It initially sets the flow on each edge to 0 and then randomly selects a path from the source s to the sink t. The maximum flow along this path is equal to the smallest capacity along this path. This flow is then added to the total flow while at the same time its value is subtracted from the capacities along this path. This procedure is repeated until no path can be found any more. It has been shown that by finding the path in each step as a shortest path from source to sink, this strategy yields an algorithm of complexity $O(|V|\,|E|^2)$ [Dinic, 1970, Edmonds and Karp, 1972].

A different approach is followed by so-called *push-relabel* algorithms [Goldberg and Tarjan, 1988]. In contrast to the Ford-Fulkerson method which looks at a complete path in the network in each step, push-relabel algorithms work locally on one vertex at a time. Furthermore, while Ford-Fulkerson type methods satisfy the flow conservation property in each step, i.e. for each node, the total incoming flow is equal to the outgoing flow, this is not the case for push-relabel algorithms. Here, the algorithm maintains a preflow, i.e. the inflow of a vertex may exceed its outflow, which is referred to as excess flow. The main idea of the method is to assign a height to each vertex. In each step a vertex with excess flow is identified and flow is sent down to a neighbor of lower height ("push"). If no such neighbor can be found, the height of the vertex needs to be increased ("relabel"). This is repeated until there is no excess flow anymore in the network. The algorithm by Goldberg and Tarjan [1988] has complexity $O(|V|\,|E| \log(|V|^2 / |E|))$. Several other algorithms have been proposed based on preflows, for instance the method by King et al. [1994], which has complexity $O(|V|\,|E| \log_{|E|/(V \log(V))}(|V|))$. The pseudoflow algorithm by Hochbaum [2008] goes one step further, as here also deficit flows are allowed during execution of the algorithm, i.e. the outflow of the vertex may exceed the inflow. Their algorithm computes a solution of the maximum flow problem in $O(|V|\,|E| \log(|E|))$.

The cut objective has been applied e.g. in clustering [Wu and Leahy, 1993] and computer vision [Boykov et al., 2001]. However, there are some drawbacks when using this clustering objective. As the value of the cut grows with the number of edges, any algorithm will tend to remove small loosely connected subsets at the boundaries of the data. This is illustrated

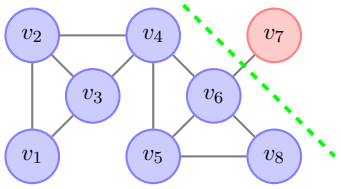

Figure 7.2: Minimizing cut leads to highly unbalanced clusters.

in Figure 7.2. It is clear that in most practical situations this is not the desired result as one would prefer the clusters to be balanced in the sense that the sizes of the two clusters do not differ too much. This leads to the balanced graph cut criteria which are discussed in the following.

7.1.2 Balanced graph cuts

In many applications one likes to avoid solutions where the size of the clusters is very unbalanced. This is achieved by *balanced graph cuts* which are discussed in this section. A balanced graph cut problem is of the form

$$\min_{C \subset V} \frac{\text{cut}(C, \overline{C})}{\widehat{S}(C)},$$

where $\hat{S}(C)$ is a symmetric balancing function. An example is the *ratio cut* [Hagen and Kahng, 1991], which is given as

$$\text{RCut}(C, \overline{C}) = \frac{\text{cut}(C, \overline{C})}{|C|} + \frac{\text{cut}(C, \overline{C})}{|\overline{C}|} = \frac{\text{cut}(C, \overline{C}) \cdot |V|}{|C| \cdot |\overline{C}|}. \tag{7.1}$$

This objective implicitly penalizes clusterings with unbalanced cluster sizes, since small clusters will lead to a small denominator. A slightly different balancing behavior is induced by the *ratio Cheeger cut* [Buser, 1978]

$$\text{RCC}(C, \overline{C}) = \frac{\text{cut}(C, \overline{C})}{\min\left\{|C|, |\overline{C}|\right\}}. \tag{7.2}$$

To see the difference in balancing behavior of the two criteria, note that we can rewrite the objectives as

$$\begin{aligned} \text{RCut}(C, \overline{C}) &= \text{cut}(C, \overline{C}) \left(\tfrac{1}{|C|} + \tfrac{1}{|\overline{C}|}\right) \qquad \text{and} \\ \text{RCC}(C, \overline{C}) &= \text{cut}(C, \overline{C}) \ \max\left\{\tfrac{1}{|C|}, \tfrac{1}{|\overline{C}|}\right\}. \end{aligned}$$

The term on the right side can be interpreted as L_1 norm of the vector consisting of the $\frac{1}{|C|}$-terms for each subset in the case of RCut, and as L_∞

norm in the case of RCC. Thus, RCC leads to a more stricter balancing behavior than RCut. This is illustrated in Figure 7.3 where we compare the balancing functions of RCut (rescaled with factor 2) and RCC. While both achieve their maximum at $\frac{1}{2}|C|$, thus favoring balanced partitions over unbalanced ones, the RCC enforces this balance more strictly than the RCut.

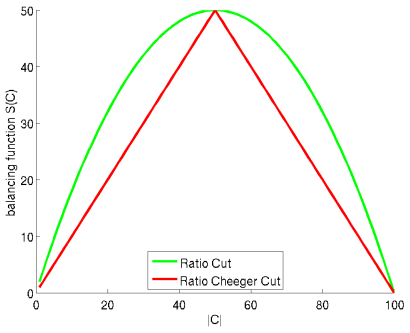

Figure 7.3: Illustration of different balancing set functions.

Another possibility is to favor clusterings where the sum of the degrees in each cluster is roughly equal. This is given by the minimizer of the *normalized cut* [Shi and Malik, 2000]:

$$\mathrm{NCut}(C,\overline{C}) = \frac{\mathrm{cut}(C,\overline{C})}{\mathrm{vol}(C)} + \frac{\mathrm{cut}(C,\overline{C})}{\mathrm{vol}(\overline{C})} = \frac{\mathrm{cut}(C,\overline{C})\cdot\mathrm{vol}(V)}{\mathrm{vol}(C)\cdot\mathrm{vol}(\overline{C})}. \tag{7.3}$$

Analogously to the ratio cut one can define a variant with a different balancing behavior, the *normalized Cheeger cut* NCC, given as [Chung, 1997]

$$\mathrm{NCC}(C,\overline{C}) = \frac{\mathrm{cut}(C,\overline{C})}{\min\left\{\mathrm{vol}(C),\mathrm{vol}(\overline{C})\right\}}. \tag{7.4}$$

The difference between RCut and NCut is illustrated in Fig. 7.4. Note that both partitions separate two edges of weight 1, i.e. $\mathrm{cut}(B,R) = 2$ in both cases, where B denotes the blue cluster and R the red cluster. However, while the partition on the left yields two clusters of equal size, i.e. $|B| = |R| = 4$, they are unbalanced with respect to the volume, as $\mathrm{vol}(B) = 24$ and $\mathrm{vol}(R) = 16$. Here, we have $\mathrm{RCut}(B,R) = 1$ and $\mathrm{NCut}(B,R) = 0.21$. In contrast to that, the partition on the right consists of two clusters of equal volume ($\mathrm{vol}(B) = \mathrm{vol}(R) = 20$) and unbalanced cluster sizes ($|B| = 3$ and $|R| = 5$). Here, we have $\mathrm{RCut}(B,R) = 1.1$ and $\mathrm{NCut}(B,R) = 0.20$. Thus RCut favors the left partition, while NCut favors the one on the right (in fact they are the optimal partitions for the corresponding criterion).

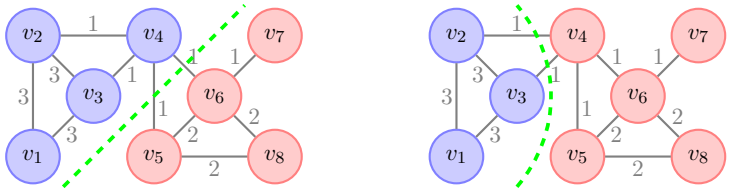

Figure 7.4: *Left:* Optimal RCut (green) *Right:* Optimal NCut (green). While RCut favors balance in size, NCut favors balance in volume.

A useful property of NCut is the following [Shi and Malik, 2000]: One can define a measure of association within clusters as

$$\mathrm{Nassoc}(C, \overline{C}) = \frac{\mathrm{assoc}(C)}{\mathrm{vol}(C)} + \frac{\mathrm{assoc}(\overline{C})}{\mathrm{vol}(\overline{C})},$$

where $\mathrm{assoc}(C) = \sum_{i,j \in C} w_{ij}$. This measure reflects how tightly nodes within clusters are connected to each other. It is now easy to see that NCut and Nassoc are connected via the simple relation $\mathrm{Nassoc}(C, \overline{C}) = 2 - \mathrm{NCut}(C, \overline{C})$, and thus it follows that minimizing NCut is equivalent to maximizing Nassoc. This connection implies that minimizing NCut enforces high intra-cluster similarity and low inter-cluster similarity simultaneously. Analogously, one shows that minimizing NCC is equivalent to maximizing

$$\mathrm{NCassoc}(C, \overline{C}) = \min \left\{ \frac{\mathrm{assoc}(C)}{\mathrm{vol}(C)}, \frac{\mathrm{assoc}(\overline{C})}{\mathrm{vol}(\overline{C})} \right\}.$$

A different way of interpreting balanced graph cuts is by random walks on a graph, which we will briefly sketch here. For a more detailed overview we refer to von Luxburg [2007]. Consider a random walk on a graph with transition probability $P = D^{-1}W$. This random walk has the stationary distribution $\pi = \frac{d}{\mathrm{vol}(V)}$, as one can easily check that $P^T\pi = \pi$. Starting at this stationary distribution π, the probability of transitioning from a set A to a set B is given as

$$\mathrm{P}(A \to B | A) = \frac{\mathrm{P}(A \to B, A)}{\mathrm{P}(A)} = \frac{\sum_{i \in A, j \in B} \pi_i P_{ij}}{\sum_{i \in A} \pi_i} = \frac{\sum_{i \in A, j \in B} w_{ij}}{\mathrm{vol}(A)}.$$

This implies that the normalized cut between two sets C and $\overline{C}$ can be written as [Meila and Shi, 2001]

$$\mathrm{NCut}(C, \overline{C}) = \frac{\mathrm{cut}(C, \overline{C})}{\mathrm{vol}(C)} + \frac{\mathrm{cut}(C, \overline{C})}{\mathrm{vol}(\overline{C})} = \mathrm{P}(C \to \overline{C} | C) + \mathrm{P}(\overline{C} \to C | \overline{C}),$$

and analogously for the normalized Cheeger cut, we obtain

$$\mathrm{NCC}(C, \overline{C}) = \max \left\{ \mathrm{P}(C \to \overline{C} | C), \mathrm{P}(\overline{C} \to C | \overline{C}) \right\}.$$

This means that when minimizing NCut/NCC, we are searching for a partition into C and $\overline{C}$ such that the random walk has a low probability of transitioning from C to $\overline{C}$ and vice versa.

This is illustrated in Fig. 7.5 for the same weighted graph used in Fig. 7.4. Here, the width as well as the grey value of the arrows is chosen proportionally to the probability $\frac{1}{d_i} w_{ij}$ of making a step from vertex i to j. One observes that a random walk starting in one of the blue vertices is more likely to stay in the blue cluster than going to one of the red vertices. For vertex v_4, switching to the blue cluster and staying in the red cluster have the same probability. However, note that the probability of stepping to vertex v_4 is higher for one of the vertices v_5 or v_6 than one of the vertices v_2 or v_3, which is why v_4 appears in the red cluster and not the blue cluster.

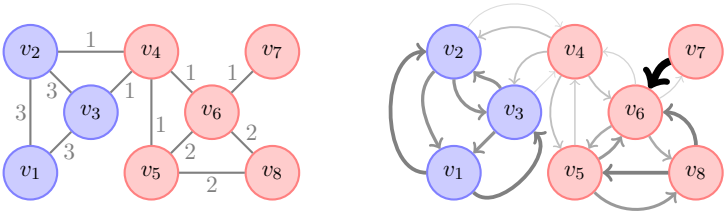

Figure 7.5: *Left:* A weighted example graph and the optimal partition according to NCut. *Right:* Random walk interpretation of NCut criterion for the example on the left. Width and grey value of arrows represent probability of stepping from vertex i to j. The random walk is more likely to stay in the blue cluster than transitioning to the red cluster.

Several other balanced graph cut criteria exist. For instance, Ding et al. [2001] considered the *min-max cut*, given as

$$\mathrm{Mcut}(C, \overline{C}) = \frac{\mathrm{cut}(C, \overline{C})}{\mathrm{assoc}(C)} + \frac{\mathrm{cut}(C, \overline{C})}{\mathrm{assoc}(\overline{C})}. \tag{7.5}$$

Other types of balancing functions where studied by Hein and Setzer [2011].

The minima of the RCC and NCC criteria in (7.2) and (7.4) are often referred to as *Cheeger constants* or *isoperimetric constants* in the literature, see e.g. Chung [1997]. The term isoperimetric arises from a geometrical motivation. The classical isoperimetric problem is given as finding, among all closed curves which have the same perimeter, the one which encloses the maximum area. Clearly in a two-dimensional plane the solution is given by the circle. Transferred to a graph-based setting, interpreting the graph as a discretization of a manifold, one is now similarly interested in relating some notion of size of the boundary of a given subset of the graph to some notion of size of the subgraph.

Interpreting the cut as size of the boundary of the set C then leads to the RCC and NCC criteria discussed above. A different way of defining the boundary of a graph subset C is as the number of vertices which are neighbors of C but not contained in C. Replacing the numerator in RCC or NCC by $|\mathrm{N}(C)|$, where $\mathrm{N}(C)$ is the set of vertices which are adjacent to some vertex in C but not contained in C, i.e. $\mathrm{N}(C) := \{j \in V \backslash C \mid \exists\, i \in C, w_{ij} > 0\}$, leads to the *vertex expansion* of the subset. Analogously, the cut based objectives are often called *edge expansion* [Chung, 1997, Hoory et al., 2006]. Moreover, note that the RCC is often simply called *expansion* in the literature, while the NCC is referred to as *conductance*, see e.g. Kannan et al. [2004]. Furthermore, the minimum of the RCut objective is often referred to as *sparsest cut*, see e.g. Leighton and Rao [1999].

Graph partitioning based on the above discussed balanced graph cuts has found applications in a diverse range of fields such as image segmentation [Shi and Malik, 2000], word-document co-clustering [Dhillon, 2001], blind source separation [Bach and Jordan, 2006], or geometry processing [Zhang et al., 2010]. Moreover, in many applications it is desirable to construct graphs with a high isoperimetric constant, since intuitively, this means that the graph is highly connected (while at the same time being sparse). These graphs, which are referred to as *expander graphs*, have applications in pseudorandom number generators or the design of communication networks, see Hoory et al. [2006] for an overview.

Note that finding the minimum of the balanced graph cuts (7.1) - (7.5) is an NP hard problem, see e.g. Shi and Malik [2000], Ding et al. [2001], Šíma and Schaeffer [2006]. Thus, typically the problem is relaxed to a tractable problem, for instance a linear program [Leighton and Rao, 1999], semi-definite program [Arora et al., 2004] or eigenproblem [von Luxburg, 2007]. In the next section we discuss the standard approach to relax the combinatorial problem to an eigenproblem involving the graph Laplacian, which will lead to the popular spectral clustering method.

7.2 Spectral clustering

In recent years, spectral clustering techniques have become one of the most popular family of clustering methods. In spectral clustering, one computes the eigenvector associated to the second eigenvalue of the graph Laplacian, a linear operator defined on the vertices of the graph. As we will show in the following, this can be motivated as a relaxation of the NP-hard problem of finding the optimal NCut or RCut on a similarity graph.

The idea of using eigenvectors for graph partitioning can be traced back to the work of Hall [1970], Donath and Hoffman [1973] and Fiedler [1973]. Later it has been rediscovered in different areas including the solution of sparse linear systems [Pothen et al., 1990], load balancing [Simon, 1991,

Hendrickson and Leland, 1995, Driessche and Roose, 1995] and circuit design [Hagen and Kahng, 1991, Chan et al., 1994]. In the machine learning community, Shi and Malik [2000] proposed the relaxation of the NCut criterion to the normalized graph Laplacian and applied it to the problem of image segmentation. The interpretation of NCut in terms of a random walk on a graph was given by Meila and Shi [2001]. A slightly different variant of the normalized graph Laplacian was used by Ng et al. [2001], and the min-max cut criterion was used by Ding et al. [2001].

Several authors performed theoretical analyses of the performance of spectral clustering. Guattery and Miller [1998] constructed several examples of graphs where spectral clustering provably leads to partitions achieving the worst-case bounds giving by the isoperimetric inequality discussed later in this section. However, Spielman and Teng [2007] proved that spectral clustering produces good partitions on several classes of graphs which often arise in practice. Worst-case guarantees for spectral clustering were given by Kannan et al. [2004] with respect to a bi-criteria measure based on the minimum NCC within each cluster and the total edge weight between clusters, and it was shown that in the presence of a "good" partition, it is found by the spectral method.

Bach and Jordan [2006] considered the problem of learning the similarity matrix used in spectral clustering from given data. Dhillon et al. [2004] showed that for a particular choice of weights, the normalized cut is equal to the kernel k-means objective. Moreover, the connection to kernel principal component analysis was demonstrated by Bengio et al. [2004]. Furthermore, the graph Laplacian has also been used in (nonlinear) dimensionality reduction [Belkin and Niyogi, 2002, Nadler et al., 2006], transductive learning [Joachims, 2003] as well as semi-supervised learning [Zhu et al., 2003, Belkin and Niyogi, 2004].

We refer to von Luxburg [2007] for an overview about various aspects of spectral clustering. In the following we will discuss the standard spectral relaxation of the NCut and RCut objectives. Note that while the performance of spectral clustering algorithms depends on the choice of the underlying graph construction (see e.g. Zelnik-Manor and Perona [2004], von Luxburg [2007], Daitch et al. [2009], Jebara et al. [2009], Maier et al. [2013]), we now consider the graph as fixed and focus on the optimization of the balanced graph cut criteria.

7.2.1 Spectral relaxation of balanced graph cuts.

In the following we review how spectral clustering can be derived as a continuous relaxation of RCut and NCut, respectively, see von Luxburg [2007].

We recall the RCut and NCut problems, given as

$$\min_{C\in V} \mathrm{RCut}(C,\overline{C}) = \min_{C\in V} \frac{\mathrm{cut}(C,\overline{C})\,|V|}{|C|\,|\overline{C}|} \quad \text{and}$$
$$\min_{C\in V} \mathrm{NCut}(C,\overline{C}) = \min_{C\in V} \frac{\mathrm{cut}(C,\overline{C})\,\mathrm{vol}(V)}{\mathrm{vol}(C)\,\mathrm{vol}(\overline{C})}.$$

The first step is to rewrite RCut and NCut as an optimization problem over $\{0,1\}^n$. Consider the functionals given by

$$Q_2^{(\mathrm{u})}(f) = \frac{R_2(f)}{S_2^{(\mathrm{u})}(f)} = \frac{\frac{1}{2}\sum_{i,j\in V} w_{ij}\,(f_i-f_j)^2}{\|f-\mathrm{mean}(f)\mathbf{1}\|_2^2} \quad \text{and}$$
$$Q_2^{(\mathrm{n})}(f) = \frac{R_2(f)}{S_2^{(\mathrm{n})}(f)} = \frac{\frac{1}{2}\sum_{i,j\in V} w_{ij}\,(f_i-f_j)^2}{\|f-\mathrm{mean}_d(f)\mathbf{1}\|_{2,d}^2}$$

where the weighted 2-norm $\|f\|_{2,d}$ is given by $\|f\|_{2,d}^2 = \sum_{i\in V} d_i\,|f_i|^2$ and the mean and its weighted variant $\mathrm{mean}_d(f)$ are given as

$$\mathrm{mean}(f) = \frac{\langle \mathbf{1}, f\rangle}{|V|} \quad \text{and} \quad \mathrm{mean}_d(f) = \frac{\langle d, f\rangle}{\mathrm{vol}(V)}.$$

We first show that $Q_2^{(\mathrm{u})}$ and $Q_2^{(\mathrm{n})}$ are extensions of the RCut and NCut criteria (see Chapter 2).

Lemma 7.1. *For any $C \subset V$, it holds that $Q_2^{(\mathrm{u})}(\mathbf{1}_C) = \mathrm{RCut}(C,\overline{C})$ and $Q_2^{(\mathrm{n})}(\mathbf{1}_C) = \mathrm{NCut}(C,\overline{C})$.*

Proof. One has for any set $C \subset V$,

$$R_2(\mathbf{1}_C) = \tfrac{1}{2}\sum_{i\in C, j\in\overline{C}} w_{ij} + \tfrac{1}{2}\sum_{i\in\overline{C}, j\in C} w_{ij} = \mathrm{cut}(C,\overline{C}),$$

where we used the symmetry of W. Using the fact that

$$S_2^{(\mathrm{n})}(f) = \big\|D^{\frac{1}{2}}(f - \tfrac{\langle d,f\rangle}{\mathrm{vol}(V)}\mathbf{1})\big\|_2^2 = \langle f, Df\rangle - \tfrac{\langle d,f\rangle^2}{\mathrm{vol}(V)},$$

one obtains

$$S_2^{(\mathrm{n})}(\mathbf{1}_C) = \mathrm{vol}(C) - \tfrac{\mathrm{vol}(C)^2}{\mathrm{vol}(V)} = \tfrac{\mathrm{vol}(C)\,\mathrm{vol}(\overline{C})}{\mathrm{vol}(V)},$$

and analogously for the unnormalized case. Thus one has $Q_1^{(\mathrm{u})}(\mathbf{1}_C) = \mathrm{RCut}(C,\overline{C})$ and $Q_2^{(\mathrm{n})}(\mathbf{1}_C) = \mathrm{NCut}(C,\overline{C})$. □

Now one can give the standard spectral relaxation of the above balanced graph cut criteria.

Theorem 7.2 (Standard spectral relaxation). *It holds that*

$$\min_{C\subset V} \mathrm{RCut}(C,\overline{C}) \;\geq\; \min_{f\in\mathbb{R}^n}\; Q_2^{(\mathrm{u})}(f) \qquad \textit{and}$$
$$\min_{C\subset V} \mathrm{NCut}(C,\overline{C}) \;\geq\; \min_{f\in\mathbb{R}^n}\; Q_2^{(\mathrm{n})}(f).$$

Proof. Lemma 7.1 implies that the original RCut and NCut problems are equivalent to the problems

$$\min_{f\in\{\mathbf{1}_C \mid C\subset V\}} Q_2^{(\mathrm{u})}(f) \qquad \text{and} \qquad \min_{f\in\{\mathbf{1}_C \mid C\subset V\}} Q_2^{(\mathrm{n})}(f),$$

respectively. We now relax the problem by replacing the constraint $f \in \{\mathbf{1}_C \mid C \subset V\}$ by $f \in \mathbb{R}^n$. As the functionals $Q_2^{(\mathrm{u})}$ and $Q_2^{(\mathrm{n})}$ are now being optimized over a superset of $\{\mathbf{1}_C \mid C \subset V\}$, one obtains a lower bound on the optimal value of RCut and NCut. □

While the functionals R, $S_2^{(\mathrm{u})}$, and $S_2^{(\mathrm{n})}$ are extensions of the corresponding set functions, they are different from their Lovász extensions. Thus, as we will see in Section 7.2.3, the relaxation is not tight. The reason for the term *spectral* will become clear in the next section, where we show that the global minimizers of the relaxed problems are given by the second eigenvectors of the unnormalized and normalized graph Laplacian, respectively.

7.2.2 Connection to eigenvectors of the graph Laplacian

We now relate the solution of the spectral relaxation of the NCut and RCut criteria to the eigenvectors of the unnormalized and normalized graph Laplacians defined below, see e.g. Mohar [1991], Chung [1997], von Luxburg [2007].

Definition 7.3 (Graph Laplacian). *The unnormalized graph Laplacian* $\Delta_2^{(\mathrm{u})}$ *and normalized graph Laplacian* $\Delta_2^{(\mathrm{n})}$ *are defined as*

$$\Delta_2^{(\mathrm{u})} = D - W$$
$$\Delta_2^{(\mathrm{n})} = D^{-1}(D - W).$$

The following Lemma shows that the unnormalized graph Laplacian is the operator which induces the quadratic form appearing in the numerator of $Q_2^{(\mathrm{u})}$ and $Q_2^{(\mathrm{n})}$ for a function $f : V \to \mathbb{R}$ via the standard inner product. Analogously, the normalized graph Laplacian is obtained for the weighted inner product, $\langle f, g\rangle_d = \sum_{i=1}^n d_i\, f_i\, g_i$.

Lemma 7.4. *The following statements hold.*

$$\tfrac{1}{2}\sum_{i,j\in V} w_{ij}(f_i - f_j)^2 \;=\; \big\langle f, \Delta_2^{(\mathrm{u})} f\big\rangle \;=\; \big\langle f, \Delta_2^{(\mathrm{n})} f\big\rangle_d,$$
$$\nabla_f\Big(\tfrac{1}{2}\sum_{i,j\in V} w_{ij}(f_i - f_j)^2\Big) \;=\; 2\,\Delta_2^{(\mathrm{u})} f \;=\; 2\,D\Delta_2^{(\mathrm{n})} f.$$

Proof. It holds that

$$\begin{aligned}\langle f, (D-W)f\rangle &= \sum_{i\in V} f_i\left((D-W)f\right)_i = \sum_{i\in V} d_i f_i^2 - \sum_{i\in V} f_i (Wf)_i \\ &= \sum_{i,j\in V} w_{ij} f_i^2 - \sum_{i,j\in V} w_{ij} f_i f_j = \tfrac{1}{2}\sum_{i,j\in V} w_{ij}(f_i - f_j)^2,\end{aligned}$$

where we have used the fact that W is symmetric. Clearly, it holds that $\left\langle f, \Delta_2^{(\mathrm{n})} f\right\rangle_d = \left\langle f, \Delta_2^{(\mathrm{u})} f\right\rangle$ as d cancels out with D^{-1}. The second statement follows directly. □

We now give an explicit characterization of the smallest eigenvalues.

Lemma 7.5. *The eigenvector of $\Delta_2^{(\mathrm{u})}$ and $\Delta_2^{(\mathrm{n})}$ corresponding to the smallest eigenvalue is the constant vector, with eigenvalue 0.*

Proof. One easily checks that $\Delta_2^{(\mathrm{u})}\mathbf{1} = 0$ and $\Delta_2^{(\mathrm{n})}\mathbf{1} = 0$ and thus $\mathbf{1}$ is an eigenvector with eigenvalue 0. Moreover, $\left\langle f, \Delta_2^{(\mathrm{u})} f\right\rangle = \frac{1}{2}\sum_{i,j=1}^n w_{ij}(f_i - f_j)^2 \geq 0$, which implies that $\Delta_2^{(\mathrm{u})}$ is positive semi-definite. Thus all eigenvalues are non-negative, and thus 0 is the smallest eigenvalue. To show the positive semi-definiteness of $\Delta_2^{(\mathrm{n})}$, we introduce the auxiliary object $\Delta_2^{(\mathrm{sym})} = D^{-\frac{1}{2}}\Delta_2^{(\mathrm{u})}D^{-\frac{1}{2}}$, which is often referred to as the symmetric graph Laplacian in the literature. Plugging the vector $f = D^{-\frac{1}{2}}g$ into the quadratic form $\left\langle f, \Delta_2^{(\mathrm{u})} f\right\rangle$ yields $0 \leq \left\langle D^{-\frac{1}{2}}g, \Delta_2^{(\mathrm{u})} D^{-\frac{1}{2}} g\right\rangle = \left\langle g, \Delta_2^{(\mathrm{sym})} g\right\rangle$, which implies that $\Delta_2^{(\mathrm{sym})}$ is positive semi-definite as well. One can now easily show that v is an eigenvector of $\Delta_2^{(\mathrm{n})}$ with eigenvalue λ if and only if $D^{\frac{1}{2}}v$ is an eigenvector of $\Delta_2^{(\mathrm{sym})}$ with the same eigenvalue. Thus $\Delta_2^{(\mathrm{n})}$ has to be positive semi-definite as well, which implies that 0 is the smallest eigenvalue. □

The following theorem shows that the solution of the spectral relaxation is given by the second eigenvector of the unnormalized graph Laplacian (we recall our assumption that the graph is connected).

Theorem 7.6 (Second eigenvalue of graph Laplacian). *The global minimum of the functional $Q_2^{(\mathrm{u})}$ is given by the second smallest eigenvalue of the unnormalized graph Laplacian $\Delta_2^{(\mathrm{u})}$, and the minimizer is the corresponding eigenvector. The corresponding statement holds for the normalized graph Laplacian $\Delta_2^{(\mathrm{n})}$.*

Proof. We simplify the denominator of $Q_2^{(\mathrm{u})}$ by applying the substitution $g := f - \mathrm{mean}(f)\mathbf{1}$, which implies that $\mathrm{mean}(g) = 0$. Rewriting this as

$\langle g, \mathbf{1} \rangle = 0$ and applying Lemma 7.4 then leads to the problem

$$\begin{aligned} \min_{g \in \mathbb{R}^n} \; & \frac{\langle g, \Delta_2^{(\mathrm{u})}\, g \rangle}{\|g\|_2^2} \\ \text{subject to } & \langle g, \mathbf{1} \rangle = 0. \end{aligned}$$

By the Rayleigh-Ritz principle (see Chapter 3), any local minimum of the above functional is given by an eigenvalue of the matrix $\Delta_2^{(\mathrm{u})}$. By Lemma 7.5, the first eigenvector is the constant vector. Due to the constraint $\langle g, \mathbf{1} \rangle = 0$ it follows that the solution of the above problem is given by the second eigenvector of the Laplacian matrix $\Delta_2^{(\mathrm{u})}$.

Similarly, for the normalized graph Laplacian, one applies the substitution $g := f - \mathrm{mean}_d(f)\mathbf{1}$, which implies that $\mathrm{mean}_d(g) = 0$. Writing this as $\langle g, d \rangle = 0$, one obtains the problem

$$\begin{aligned} \min_{g \in \mathbb{R}^n} \; & \frac{\langle g, \Delta_2^{(\mathrm{u})}\, g \rangle}{\langle g, Dg \rangle} \\ \text{subject to } & \langle g, D\mathbf{1} \rangle = 0. \end{aligned}$$

Again substituting $g = D^{-\frac{1}{2}} h$, one obtains

$$\begin{aligned} \min_{h \in \mathbb{R}^n} \; & \frac{\langle h, D^{-\frac{1}{2}} \Delta_2^{(\mathrm{u})} D^{-\frac{1}{2}}\, h \rangle}{\|h\|_2^2} \\ \text{subject to } & \langle h, D^{\frac{1}{2}} \mathbf{1} \rangle = 0. \end{aligned}$$

One now uses the fact that v is an eigenvector of $\Delta_2^{(\mathrm{n})}$ with eigenvalue λ if and only if $D^{\frac{1}{2}} v$ is an eigenvector of $\Delta_2^{(\mathrm{sym})} = D^{-\frac{1}{2}} \Delta_2^{(\mathrm{u})} D^{-\frac{1}{2}}$ with the same eigenvalue. Thus Lemma 7.5 implies that $D^{\frac{1}{2}}\mathbf{1}$ is the smallest eigenvector of $\Delta_2^{(\mathrm{sym})}$. By the Rayleigh-Ritz principle we now obtain that the minimizer h^* of the above problem is the second eigenvector of $\Delta_2^{(\mathrm{sym})}$, and the minimizer $g^* = D^{-\frac{1}{2}} h^*$ of the original problem is the second eigenvector of $\Delta_2^{(\mathrm{n})}$. □

The above result suggest that the optimal value of the spectral relaxation can be obtained by computing the second eigenvector of the corresponding graph Laplacian $\Delta_2^{(\mathrm{u})}$ or $\Delta_2^{(\mathrm{n})}$. To achieve this, standard techniques for eigenvector computation such as the inverse power method (see Chapter 5) can be used. The real-valued solution of these eigenproblems can then be transformed back into a partition of the graph via thresholding. The optimal threshold is found by optimizing the original RCut or NCut criterion, i.e. for RCut one solves

$$C' = \operatorname*{arg\,min}_{C_i, i=1,\ldots,n} \mathrm{RCut}(C_i, \overline{C}_i),$$

where the sets C_i are defined as $C_i := \{j \in V | f_j \geq f_i\}$ for $i = 1, \ldots, n$. One can now derive upper and lower bounds on the second eigenvalue in terms of the optimal cut, which we will discuss in the next section.

7.2.3 Isoperimetric inequality for spectral relaxation

The *isoperimetric inequality* [Cheeger, 1970, Dodziuk, 1984, Alon and Milman, 1985] for the graph Laplacian provides additional theoretical backup for the spectral relaxation. It provides upper and lower bounds on the RCC and NCC in terms of the second eigenvalue of the graph Laplacian. Let us now introduce the notation h_{RCC} and h_{NCC} for the optimal ratio and normalized Cheeger cut values, given as

$$h_{\mathrm{RCC}} = \inf_{C \subset V} \mathrm{RCC}(C, \overline{C}) \text{ and } h_{\mathrm{NCC}} = \inf_{C \subset V} \mathrm{NCC}(C, \overline{C}).$$

Cheeger's inequality relates the second smallest eigenvalue of the Laplacian to the isoperimetric constants defined above. Based on an analogous result for Riemannian manifolds by Cheeger [1970], the statement for graphs can be traced back to Dodziuk [1984] and Alon and Milman [1985]. The standard Cheeger isoperimetric inequality (see also Chung [1997]) is given as follows.

Theorem 7.7 (Cheeger's inequality). *Denote by λ_2 the second eigenvalue of the normalized graph Laplacian $\Delta_2^{(\mathrm{n})}$. Then,*

$$\frac{h_{\mathrm{NCC}}^2}{2} \leq \lambda_2 \leq 2\, h_{\mathrm{NCC}},$$

Denote by λ_2 the second eigenvalue of the unnormalized graph Laplacian $\Delta_2^{(\mathrm{u})}$. Then,

$$\frac{h_{\mathrm{RCC}}^2}{2 \max_{i \in V} d_i} \leq \lambda_2 \leq 2 h_{\mathrm{RCC}} .$$

Generalizations to higher order eigenvalues were considered by Chung et al. [2000], Daneshgar et al. [2010]. One can now establish a connection between the optimal Cheeger cut and the cut which is obtained by performing optimal thresholding of the second eigenvector of the graph Laplacian according to the NCC or RCC criterion. For a proof see [Bühler and Hein, 2009b].

Theorem 7.8 (Bounds on obtained cuts). *Let h_{RCC}^* denote the* RCC *value obtained by optimal thresholding of the second eigenvector of the unnormalized graph Laplacian $\Delta_2^{(\mathrm{u})}$. Then*

$$h_{\mathrm{RCC}} \leq h_{\mathrm{RCC}}^* \leq 2 \left(\max_{i \in V} d_i \right)^{\frac{1}{2}} \left(h_{\mathrm{RCC}} \right)^{\frac{1}{2}} .$$

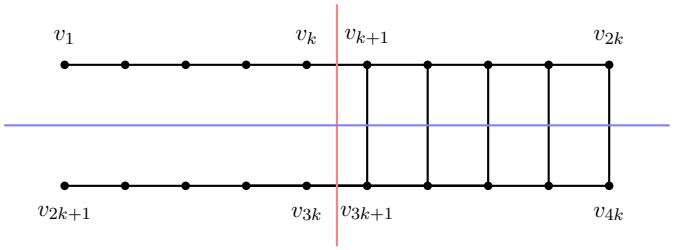

Figure 7.6: The cockroach graph considered by Guattery and Miller [1998], von Luxburg [2007]. Optimal cut (red) and cut found by spectral clustering (blue).

Let h^*_{NCC} *denote the* NCC *value obtained by optimal thresholding of the second eigenvector of the normalized graph Laplacian* $\Delta_2^{(\mathrm{n})}$*. Then,*

$$h_{\mathrm{NCC}} \;\leq\; h^*_{\mathrm{NCC}} \;\leq\; 2\left(h_{\mathrm{NCC}}\right)^{\frac{1}{2}} .$$

An advantage of the spectral relaxation is that it leads to a standard problem from linear algebra, which can be solved efficiently. However, while the above result gives a certain worst-case guarantee on the quality of the relaxation, the bound is still quite loose, and the relaxation can lead to a solution far away from the optimum.

As an example consider the cockroach or ladder graph considered by Guattery and Miller [1998] (see also the discussion in von Luxburg [2007]). These graphs look like a cockroach, or a ladder with some missing rungs (see Fig. 7.6). The optimal RCut (shown in red) cuts the graph vertically such that $C = (v_1, \ldots, v_k, v_{2k+1}, \ldots v_{3k})$ and $\overline{C} = (v_{k+1}, \ldots, v_{2k}, v_{3k+1}, \ldots, v_{4k})$. Here we have $\mathrm{cut}(C, \overline{C}) = 2$ and $|C| = \left|\overline{C}\right| = 2k$ and hence $\mathrm{RCut}(C, \overline{C}) = \frac{2}{k}$. However, as shown in Guattery and Miller [1998], unnormalized spectral clustering leads to a horizontal cut (shown in blue), which partitions the graph into sets $C = (v_1, \ldots, v_{2k})$ and $\overline{C} = (v_{2k+1}, \ldots, \ldots, v_{4k})$. Here we have $\mathrm{cut}(C, \overline{C}) = k$ and $|C| = \left|\overline{C}\right| = 2k$ and hence $\mathrm{RCut}(C, \overline{C}) = 1$. Hence in this example, spectral clustering leads to a cut which is a factor $\frac{2}{k}$ worse than the optimal cut.

In the next section, we will show that better bounds can be obtained for the eigenvalues of the so-called graph p-Laplacian.

7.3 p-Spectral clustering

p-Spectral clustering was introduced in Bühler [2009], Bühler and Hein [2009a] as a generalization of standard spectral clustering. The main motivation for p-spectral clustering is that while standard spectral clustering

corresponds to an optimization problem involving quadratic functionals, it may pose an advantage to go over to a more general setting where the quadratic functionals are replaced by functionals of a power p, for $p < 2$. Similar ideas were used to develop methods for semi-supervised learning [Zhou and Schölkopf, 2005] and image processing [Elmoataz et al., 2008]. Moreover, different generalizations of the notion of resistance between two vertices in a graph were studied by Herbster and Lever [2009] as well as Alamgir and von Luxburg [2011].

To derive p-spectral clustering, one can proceed similar to the standard spectral relaxation of the NCut criterion. First one finds an extension of the combinatorial objective to the continuous domain, by rewriting the combinatorial problem as an optimization problem over $\{0,1\}^n$. This problem is then relaxed to an optimization problem over $\mathbb{R}^n$. In contrast to the standard relaxation, the resulting optimization problem does not lead to a standard linear eigenproblem, but instead a nonlinear eigenproblem involving the unnormalized and normalized graph p-Laplacian $\Delta_p^{(\mathrm{u})}$ and $\Delta_p^{(\mathrm{n})}$, which will be defined in this section. By extending a result by Amghibech [2003], we will later show that this leads to tighter bounds on the obtained Cheeger cut values than the standard spectral relaxation.

7.3.1 p-Spectral relaxation of balanced graph cuts

We introduce the following two classes of balanced graph cuts.

$$\mathrm{RCC}_p(C,\overline{C}) = \mathrm{cut}(C,\overline{C})\ \frac{\left(|\overline{C}|^{\frac{1}{p-1}} + |C|^{\frac{1}{p-1}}\right)^{p-1}}{|C||\overline{C}|} \quad \text{and}$$

$$\mathrm{NCC}_p(C,\overline{C}) = \mathrm{cut}(C,\overline{C})\ \frac{\left(\mathrm{vol}(\overline{C})^{\frac{1}{p-1}} + \mathrm{vol}(C)^{\frac{1}{p-1}}\right)^{p-1}}{\mathrm{vol}(C)\,\mathrm{vol}(\overline{C})}.$$

The following proposition shows that the above balanced graph cut criteria are generalizations of the graph cut criteria discussed before.

Proposition 7.9. *It holds that* $\lim_{p\to 1}\mathrm{NCC}_p(C,\overline{C}) = \mathrm{NCC}(C,\overline{C})$ *as well as* $\mathrm{NCC}_2(C,\overline{C}) = \mathrm{NCut}(C,\overline{C})$. *Moreover, for any* $1 < p < 2$ *it holds that* $\mathrm{NCC}(C,\overline{C}) \leq \mathrm{NCC}_p(C,\overline{C}) \leq \mathrm{NCut}(C,\overline{C})$. *The analogous results hold for* RCC_p.

Proof. The statements for $p = 2$ can be directly seen by plugging in $p = 2$ and using that $|C| + |\overline{C}| = |V|$ as well as $\mathrm{vol}(C) + \mathrm{vol}(\overline{C}) = \mathrm{vol}(V)$. For the case $p \to 1$, note that one has $\lim_{\alpha\to\infty}(a^\alpha + b^\alpha)^{\frac{1}{\alpha}} = \max\{a,b\}$ for $a,b \geq 0$. Therefore one obtains

$$\lim_{p\to 1}\left(\mathrm{vol}(\overline{C})^{\frac{1}{p-1}} + \mathrm{vol}(C)^{\frac{1}{p-1}}\right)^{p-1} = \max\{\mathrm{vol}(\overline{C}), \mathrm{vol}(C)\},$$

which implies $\lim_{p\to 1} \mathrm{NCC}_p(C,\overline{C}) = \mathrm{NCC}(C,\overline{C})$. Analogously one shows the result for RCC_p. To understand the result in the interval $1 < p < 2$, let us consider the following inequalities between l_p-norms: For $\infty \geq \alpha \geq 1$ one has $\|x\|_\infty \leq \|x\|_\alpha \leq \|x\|_1$. From this it follows that

$$\max\{\mathrm{vol}(\overline{C}), \mathrm{vol}(C)\} \leq \left(\mathrm{vol}(\overline{C})^\alpha + \mathrm{vol}(C)^\alpha\right)^{\frac{1}{\alpha}} \leq \mathrm{vol}(\overline{C}) + \mathrm{vol}(C),$$

where $\alpha = \frac{1}{p-1}$. This implies $\mathrm{NCC}(C,\overline{C}) \leq \mathrm{NCC}_p(C,\overline{C}) \leq \mathrm{NCut}(C,\overline{C})$. Analogously one shows the result for RCC_p. □

Thus, in the interval $1 < p < 2$, the RCC_p can be seen as an interpolation between RCC and RCut, and analogously in the normalized case. Let us now introduce the following functionals for the RCC_p and NCC_p criteria:

$$Q_p^{(\mathrm{u})}(f) = \frac{R_p(f)}{S_p^{(\mathrm{u})}(f)} = \frac{\frac{1}{2}\sum_{i,j\in V} w_{ij}\,|f_i - f_j|^p}{\|f - \mathrm{mean}_p(f)\mathbf{1}\|_p^p} \quad \text{and}$$

$$Q_p^{(\mathrm{n})}(f) = \frac{R_p(f)}{S_p^{(\mathrm{n})}(f)} = \frac{\frac{1}{2}\sum_{i,j\in V} w_{ij}\,|f_i - f_j|^p}{\|f - \mathrm{mean}_{p,d}(f)\mathbf{1}\|_{p,d}^p}$$

where the weighted p-norm $\|f\|_{p,d}$ is given as $\|f\|_{p,d}^p := \sum_{i\in V} d_i\,|f_i|^p$ and the p-mean and its weighted variant are defined as

$$\mathrm{mean}_p(f) = \operatorname*{arg\,min}_{m\in\mathbb{R}} \|f - m\mathbf{1}\|_p \quad \text{and} \quad \mathrm{mean}_{p,d}(f) = \operatorname*{arg\,min}_{m\in\mathbb{R}} \|f - m\mathbf{1}\|_{p,d},$$

generalizing $\mathrm{mean}(f)$ and $\mathrm{mean}_d(f)$ from the last section. Using the above functionals, we will later derive a relaxation of the balanced graph cut criteria RCC_p and NCC_p. Moreover, we introduce the notation $\phi_p(x) := |x|^{p-2}\,x$.

We first consider the differentiability of the weighted p-norm $\|f\|_{p,d}^p$. Note that while it is a well-known fact that for any $p > 1$ the p-norm is differentiable everywhere except at the origin, adding the additional power p makes the functional differentiable everywhere, as stated in the following lemma.

Lemma 7.10. *Let $p > 1$ and $d \in \mathbb{R}^n_+$. Then the function $\|f\|_{p,d}^p$ is differentiable for all $f \in \mathbb{R}^n$ and it holds that $\frac{\partial}{\partial f_k}\left(\|f\|_{p,d}^p\right) = p\,d_k\,\phi_p(f^k)$.*

Proof. Note that in order to prove that $\|f\|_{p,d}^p$ is differentiable, it is sufficient to prove that all the partial derivatives exist and are continuous for all $g \in \mathbb{R}^n$ (see e.g. Theorem 1.9.5 in Hubbard and Hubbard [1998]). For any $k \in \{1,\ldots,n\}$, one easily checks that for any $g \in \mathbb{R}^n$ with $g_k \neq 0$, the k-th partial derivative of the functional $\|f\|_{p,d}^p$ evaluated at the point g is given by

$$\frac{\partial}{\partial f_k}\left(\|f\|_{p,d}^p\right)(g) = p\,d_k\;|g_k|^{p-1}\,\mathrm{sign}(g^k) = p\,d_k\,\phi_p(g_k),$$

which is continuous at any point $g \in \mathbb{R}^n$ with $g_k \neq 0$. For the case when $g_k = 0$, we need to show that the limit

$$\lim_{h \to 0} \frac{\|(g_1, \ldots, g_k + h, \ldots, g_n)\|_{p,d}^p - \|(g_1, \ldots, g_k, \ldots, g_n)\|_{p,d}^p}{h}$$

exists. We first compute the one-sided limit

$$\begin{aligned} &\lim_{h \searrow 0} \frac{\left(\sum_{i \neq k} d_i \, |g_i|^p + d_k \, |h|^p\right) - \left(\sum_{i \neq k} d_i \, |g_i|^p\right)}{h} \\ &= \lim_{h \searrow 0} d_k \frac{|h|^p}{h} = \lim_{h \searrow 0} d_k \, |h|^{p-1} \operatorname{sign}(h) = 0, \end{aligned}$$

and similarly $\lim_{h \nearrow 0} d_k \, |h|^{p-1} \operatorname{sign}(h) = 0$, which implies that the partial derivative exists also if $g_k = 0$ and is given by

$$\frac{\partial}{\partial f_k} \left(\|f\|_{p,d}^p \right)(g) = 0 = p \, d_k \, \phi_p(g^k).$$

What is left to show is that the k-th partial derivative is continuous at every point $g \in \mathbb{R}^n$ with $g_k = 0$. Thus we need to show that

$$\lim_{(x_1, \ldots, x_k, \ldots, x_n) \to (g_1, \ldots, 0, \ldots g_n)} \frac{\partial}{\partial f_k} \left(\|f\|_{p,d}^p \right)(x) = \frac{\partial}{\partial f_k} \left(\|f\|_{p,d}^p \right)(g) \, .$$

To show this, one needs to take into account all paths $(x_1, \ldots, x_k, \ldots, x_n) \to (g_1, \ldots, 0, \ldots, g_n)$. However, note that the value of the k-th partial derivative does not depend on the values of the components at index $i \neq k$. Thus we only need to consider the k-th index and obtain

$$\lim_{(x_1, \ldots, x_k, \ldots, x_n) \to (g_1, \ldots, 0, \ldots, g_n)} \frac{\partial}{\partial f_k} \left(\|f\|_{p,d}^p \right)(x) = \lim_{x_k \to 0} p \, d_k \, \phi(x_k).$$

One easily checks that one obtains left and right one-sided limits 0. Since this agrees with the value of the derivative at g, we obtain the continuity of the k-th partial derivative, which implies that $\|f\|_{p,d}^p$ is differentiable. □

Let us now give an explicit characterization of the weighted and unweighted p-mean evaluated at indicator functions.

Lemma 7.11. *Let $p > 1$ and $C \subset V$. Then it holds that*

$$\operatorname{mean}_p(\mathbf{1}_C) = \frac{|C|^{\frac{1}{p-1}}}{|C|^{\frac{1}{p-1}} + |\overline{C}|^{\frac{1}{p-1}}} \text{ and } \operatorname{mean}_{p,d}(\mathbf{1}_C) = \frac{\operatorname{vol}(C)^{\frac{1}{p-1}}}{\operatorname{vol}(C)^{\frac{1}{p-1}} + \operatorname{vol}(\overline{C})^{\frac{1}{p-1}}}.$$

Proof. We give the proof for the weighted p-mean. The result for the p-mean follows analogously. Note that the functional $\|f - m\mathbf{1}\|_{p,d}^p$ (with the power p) is differentiable if $p > 1$, and it holds that

$$\frac{\partial}{\partial m}\Big(\|f - m\mathbf{1}\|_{p,d}^p \Big) \;=\; -p \sum_{i \in V} d_i \; \phi_p\big(f_i - m\big).$$

Thus at the optimal value $\widehat{m}$ we must have $\sum_{i \in V} d_i \; \phi_p\big(f_i - \widehat{m}\big) = 0$. Setting $f = \mathbf{1}_C$, we obtain (note that $0 \leq \widehat{m} \leq 1$),

$$0 = \sum_{i \in C} d_i \phi_p\big(1 - \widehat{m}\big) + \sum_{i \in \overline{C}} d_i \phi_p\big(-\widehat{m}\big) = \mathrm{vol}(C) \; \phi_p\big(1 - \widehat{m}\big) - \mathrm{vol}(\overline{C}) \, \phi_p\big(\widehat{m}\big).$$

For $p > 1$ one has $\widehat{m} = 0$ if and only if $\mathrm{vol}(C) = 0$, and $\widehat{m} = 1$ if and only if $\mathrm{vol}(\overline{C}) = 0$. Otherwise we have $0 < \widehat{m} < 1$ and obtain by rearranging,

$$\left(\frac{1 - \widehat{m}}{\widehat{m}}\right)^{p-1} = \frac{\mathrm{vol}(\overline{C})}{\mathrm{vol}(C)}. \qquad \Leftrightarrow \qquad \frac{1}{\widehat{m}} = \left(\frac{\mathrm{vol}(\overline{C})}{\mathrm{vol}(C)}\right)^{\frac{1}{p-1}} + 1.$$

Finally, solving for $\widehat{m}$ yields the result. □

The following lemma shows that the functionals $Q_p^{(\mathrm{u})}(f)$ and $Q_p^{(\mathrm{n})}(f)$ can be seen as extensions of the above class of balanced graph cuts. Note that this is a generalization of Lemma 7.1 from Section 7.2.

Lemma 7.12. *For every $C \subset V$ and $p > 1$, it holds that $Q_p^{(\mathrm{u})}(\mathbf{1}_C) = \mathrm{RCC}_p(C, \overline{C})$ and $Q_p^{(\mathrm{n})}(\mathbf{1}_C) = \mathrm{NCC}_p(C, \overline{C})$.*

Proof. Again we show the result only for the normalized case, the result for the unnormalized case follows analogously. It holds that

$$R_p(\mathbf{1}_C) = \tfrac{1}{2} \sum_{i \in C, j \in \overline{C}} w_{ij} + \tfrac{1}{2} \sum_{i \in \overline{C}, j \in C} w_{ij} = \sum_{i \in C, j \in \overline{C}} w_{ij} = \mathrm{cut}(C, \overline{C}),$$

where we used the symmetry of W. The denominator $S_p^{(\mathrm{n})}(\mathbf{1}_C)$ is equal to

$$\sum_{i \in C} d_i \left| 1 - \frac{\mathrm{vol}(C)^{\frac{1}{p-1}}}{\mathrm{vol}(C)^{\frac{1}{p-1}} + \mathrm{vol}(\overline{C})^{\frac{1}{p-1}}} \right|^p + \sum_{i \in \overline{C}} d_i \left| \frac{\mathrm{vol}(C)^{\frac{1}{p-1}}}{\mathrm{vol}(C)^{\frac{1}{p-1}} + \mathrm{vol}(\overline{C})^{\frac{1}{p-1}}} \right|^p,$$

where we have used Lemma 7.11. We can further reformulate this as

$$\frac{\mathrm{vol}(C)\,\mathrm{vol}(\overline{C})^{\frac{p}{p-1}} + \mathrm{vol}(\overline{C})\,\mathrm{vol}(C)^{\frac{p}{p-1}}}{\big(\mathrm{vol}(C)^{\frac{1}{p-1}} + \mathrm{vol}(\overline{C})^{\frac{1}{p-1}}\big)^p} = \frac{\mathrm{vol}(C)\,\mathrm{vol}(\overline{C})}{\big(\mathrm{vol}(C)^{\frac{1}{p-1}} + \mathrm{vol}(\overline{C})^{\frac{1}{p-1}}\big)^{p-1}},$$

which concludes the proof. □

We can now use the previous lemma to give the p-spectral relaxation of the above balanced graph cuts.

Theorem 7.13 (p-spectral relaxation). *Let $p > 1$. Then it holds that*

$$\min_{C \subset V} \mathrm{RCC}_p(C, \overline{C}) \;\geq\; \min_{f \in \mathbb{R}^n} \; Q_p^{(\mathrm{u})}(f) \qquad \textit{and}$$
$$\min_{C \subset V} \mathrm{NCC}_p(C, \overline{C}) \;\geq\; \min_{f \in \mathbb{R}^n} \; Q_p^{(\mathrm{n})}(f).$$

Proof. Lemma 7.12 implies that the original RCC_p and NCC_p problems are equivalent to the problems

$$\min_{f \in \{\mathbf{1}_C \;|\; C \subset V\}} Q_p^{(\mathrm{u})}(f) \qquad \text{and} \qquad \min_{f \in \{\mathbf{1}_C \;|\; C \subset V\}} Q_p^{(\mathrm{n})}(f),$$

respectively. Again, we now relax the problem by replacing the constraint that $f \in \{\mathbf{1}_C \mid C \subset V\}$ by $f \in \mathbb{R}^n$, yielding a lower bound on the optimal values RCC_p and NCC_p. □

Note that in Bühler [2009], a slightly different version of Theorem 7.13 was given. There is was shown that for every $C \subset V$, there exist a function $f_{p,C}^{(\mathrm{u})}$ such that $Q_p^{(\mathrm{u})}(f_{p,C}^{(\mathrm{u})}) = \mathrm{RCC}_p(C, \overline{C})$. Explicitly, $f_{p,C}^{(\mathrm{u})}$ was given as

$$(f_{p,C}^{(\mathrm{u})})_i = \begin{cases} \dfrac{1}{|C|^{\frac{1}{p-1}}}, & i \in C, \\ -\dfrac{1}{\left|\overline{C}\right|^{\frac{1}{p-1}}}, & i \in \overline{C}, \end{cases}$$

and analogously a vector $f_{p,C}^{(\mathrm{n})}$ for $Q_p^{(\mathrm{n})}$. This implies that finding a set C which minimizes the above balanced graph cut criteria is equivalent to optimizing the corresponding functional $Q_p^{(\mathrm{u})}$ or $Q_p^{(\mathrm{n})}$ over the sets $\{f_{p,C}^{(\mathrm{u})} \mid C \subset V\}$ or $\{f_{p,C}^{(\mathrm{n})} \mid C \subset V\}$, respectively. Thus, similarly to Theorem 7.13 this implies that $Q_p^{(\mathrm{u})}$ is an extension of $\mathrm{RCC}_p(C, \overline{C})$. The motivation of choosing the function $f_{p,C}^{(\mathrm{u})}$ in Bühler and Hein [2009a] was because it satisfies $\mathrm{mean}_p(f_{p,C}^{(\mathrm{u})}) = 0$. The connection to the statement from Theorem 7.13 can be seen by noting that the functional $Q_p^{(\mathrm{u})}$ satisfies $Q_p^{(\mathrm{u})}(\alpha f + \beta \mathbf{1}) = Q_p^{(\mathrm{u})}(f)$ for all $f \in \mathbb{R}^n, \alpha > 0$ and $\beta \in \mathbb{R}$, due to the p-homogeneity of numerator and denominator as well as the fact that they are invariant under addition of a constant. The equivalence between the statement in Theorem 7.13 and the corresponding result in Bühler and Hein [2009a] follows by noting that

$$f_{p,C}^{(\mathrm{u})} = \alpha \mathbf{1}_C - \mathrm{mean}_p(\alpha \mathbf{1}_C)\mathbf{1}, \qquad \text{where} \qquad \alpha = \left(\tfrac{1}{|C|}\right)^{\frac{1}{p-1}} + \left(\tfrac{1}{|\overline{C}|}\right)^{\frac{1}{p-1}}.$$

In the next section we will show that letting $p \to 1$ will lead to improving bounds in terms of the optimal RCC and NCC values. In the following section we will show the connection to the eigenvectors of the unnormalized and normalized graph p-Laplacian $\Delta_p^{(\mathrm{u})}$ and $\Delta_p^{(\mathrm{n})}$, respectively.

7.3.2 Connection to eigenvectors of the graph p-Laplacian

In this section we study the graph p-Laplacian and its associated eigenproblems and show the connection to the optimal value of the p-spectral relaxation of the balanced graph cut criteria from the last section. The unnormalized and normalized graph p-Laplacians can be defined as follows, see e.g. Holopainen and Soardi [1997], Amghibech [2003, 2006], Mugnolo [2013].

Definition 7.14 (Graph p-Laplacian). *Let $i \in V$. Then for $p > 1$, the unnormalized graph p-Laplacian $\Delta_p^{(\mathrm{u})}$ and normalized graph p-Laplacian $\Delta_p^{(\mathrm{n})}$ are defined as*

$$(\Delta_p^{(\mathrm{u})} f)_i = \sum_{j \in V} w_{ij}\, \phi_p\,(f_i - f_j),$$
$$(\Delta_p^{(\mathrm{n})} f)_i = \tfrac{1}{d_i} \sum_{j \in V} w_{ij}\, \phi_p\,(f_i - f_j),$$

where $\phi_p : \mathbb{R} \to \mathbb{R}$ is defined for $x \in \mathbb{R}$ as $\phi_p(x) = |x|^{p-2}\, x$.

The graph p-Laplacian induces the functional $R_p(f) = \frac{1}{2} \sum_{i,j \in V} |f_i - f_j|^p$ via the standard inner product in the unnormalized case, and the weighted inner product, $\langle x, y \rangle_d = \sum_{i \in V} d_i x_i y_i$, in the normalized case, as we will show in the sequel. Moreover, we give the relation to the gradient of R_p.

Lemma 7.15. *The following statements hold.*

$$\tfrac{1}{2} \sum_{i,j \in V} |f_i - f_j|^p = \left\langle f, \Delta_p^{(\mathrm{u})} f \right\rangle = \left\langle f, \Delta_p^{(\mathrm{n})} f \right\rangle_d$$
$$\nabla_f \Big(\tfrac{1}{2} \sum_{i,j \in V} |f_i - f_j|^p \Big) = \; p\, \Delta_p^{(\mathrm{u})}(f) \; = \; p\, D\, \Delta_p^{(\mathrm{n})}(f).$$

Proof. For the first statement, note that

$$\sum_{i,j \in V} w_{ij} \phi_p(f_i - f_j) f_i \; = \; \tfrac{1}{2} \sum_{i,j \in V} w_{ij} \phi_p(f_i - f_j) f_i + \tfrac{1}{2} \sum_{i,j \in V} w_{ij} \phi_p(f_j - f_i) f_j$$
$$= \tfrac{1}{2} \sum_{i,j \in V} w_{ij} \phi_p(f_i - f_j)(f_i - f_j) \; = \; \tfrac{1}{2} \sum_{i,j \in V} |f_i - f_j|^p \,,$$

where we have used the symmetry of W. Similarly one proceeds for the normalized case. For the second statement, note that by Lemma 7.10, the functional $\|f\|_{p,d}^p$ is differentiable for all $f \in \mathbb{R}^n, d \in \mathbb{R}^n_+$ if $p > 1$. By the chain rule, also R_p is differentiable and we obtain

$$\tfrac{\partial}{\partial f_k} \Big(R_p(f) \Big) = \tfrac{1}{2} \sum_{i,j \in V} p\, w_{ij} \phi_p\,(f_i - f_j) \Big(\delta_{i=k} - \delta_{j=k} \Big)$$
$$= p\, \Delta_p^{(\mathrm{u})}(f)_k \; = \; p\, d_k\, \Delta_p^{(\mathrm{n})}(f)_k,$$

which concludes the proof. □

For $p = 2$, one obtains the statements from Lemma 7.4 as special case. The eigenvectors and eigenvalues of the p-Laplacian are defined as follows.

Definition 7.16 (Eigenvalues of graph p-Laplacian). *The real number λ_p is called an eigenvalue for the normalized p-Laplacian $\Delta_p^{(\mathrm{n})}$ if there exists a vector $f \in \mathbb{R}^n$ such that*

$$(\Delta_p^{(\mathrm{n})} f)_i = \lambda_p \, \phi_p(f_i), \quad \forall\, i = 1, ..., n.$$

The vector f is called an eigenvector of $\Delta_p^{(\mathrm{n})}$.

The definition of the eigenvalues of the unnormalized graph p-Laplacian $\Delta_p^{(\mathrm{u})}$ works analogously. Again, one recovers the standard linear eigenproblems as special cases for $p = 2$. To see the origin of this definition, now consider the following functionals associated to the graph p-Laplacians $\Delta_p^{(\mathrm{u})}$ and $\Delta_p^{(\mathrm{n})}$,

$$\overline{Q}_p^{(\mathrm{u})}(f) = \frac{\frac{1}{2}\sum_{i,j=1}^n w_{ij}|f_i - f_j|^p}{\|f\|_p^p} \quad \text{and} \quad \overline{Q}_p^{(\mathrm{n})}(f) = \frac{\frac{1}{2}\sum_{i,j=1}^n w_{ij}|f_i - f_j|^p}{\|f\|_{p,d}^p}.$$

By Theorem 3.6 a necessary condition for a critical point of $\overline{Q}_p^{(\mathrm{n})}$ is given by

$$0 \in \partial\Big(\tfrac{1}{2}\sum_{i,j\in V} w_{ij}\,|f_i - f_j|^p\Big) - \lambda\,\partial\Big(\,\|f\|_{p,d}^p\Big),$$

where $\lambda = \overline{Q}_p^{(\mathrm{n})}(f)$. Note that while in general, nonlinear eigenproblems yield only a necessary condition for a critical point, in fact in this case it is also sufficient, as we will see below. Evaluating the subdifferentials will then lead to the nonlinear eigenproblem for $\Delta_p^{(\mathrm{n})}$ as defined in Def. 7.16. Analogously, one proceeds for the unnormalized case. The following proposition formalizes the connection between critical points of the functional $\overline{Q}_p^{(\mathrm{n})}$ and eigenvectors of the normalized graph p-Laplacian, see Amghibech [2003].

Proposition 7.17. *Let $p > 1$. The functional $\overline{Q}_p^{(\mathrm{n})}$ has a critical point at $f \in \mathbb{R}^n$ if and only if f is an eigenvector of $\Delta_p^{(\mathrm{n})}$. The corresponding eigenvalue λ_p is given as $\lambda_p = \overline{Q}_p^{(\mathrm{n})}(f)$.*

Proof. By Theorem 3.6 every critical point g of $\overline{Q}_p^{(\mathrm{n})}$ is a solution of the nonlinear eigenproblem

$$0 \in \partial\Big(\tfrac{1}{2}\sum_{i,j\in V} w_{ij}\,|g_i - g_j|^p\Big) - \lambda\,\partial\Big(\,\|g\|_{p,d}^p\Big)\,,$$

for $\lambda = \overline{Q}_p^{(\mathrm{n})}(g)$. Moreover, in this case this is also a sufficient condition, since for $p > 1$ the denominator is differentiable, see Lemma 7.10. The

derivative is given as $\frac{\partial}{\partial f_k}\big(\|f\|_{p,d}^p\big) = p\,d_k\,\phi_p\big(f_k\big)$. Thus with Lemma 7.15 one obtains

$$0 = p\,d_k(\Delta_p^{(\mathrm{n})} f)_k - \lambda\, p\,d_k\,\phi_p\big(f_k\big).$$

Dividing by $p\,d_k$ gives the result. Since by Theorem 3.6 every eigenvector f with eigenvalue λ satisfies $\lambda = \overline{Q}_p^{(\mathrm{n})}(f)$, also the reverse direction holds. □

An analogous statement can be made in the case of the unnormalized graph p-Laplacian. Similarly to the case $p = 2$, one can give an explicit characterization of the first eigenvector. Note that again we use the assumption that the graph G is connected.

Lemma 7.18. *Let $p > 1$. The eigenvector of $\Delta_p^{(\mathrm{u})}$ and $\Delta_p^{(\mathrm{n})}$ corresponding to the smallest eigenvalue is the constant vector, with eigenvalue 0.*

Proof. One can easily check that the nonlinear eigenproblem is fulfilled in both cases by setting $v = \mathbf{1}$ and $\lambda = 0$. Moreover, by Prop. 7.17 and the non-negativity of $\overline{Q}_p^{(\mathrm{n})}$ it follows that it has to be the smallest eigenvalue. Analogously one proceeds for the unnormalized case. □

We now go back to the relaxation of the pNCC criterion, where we optimize the function $Q_p^{(\mathrm{n})}$ introduced in the last section over $\mathbb{R}^n$. The following theorem shows that the solution of the relaxed problem is obtained by the nonlinear eigenvector corresponding to the second smallest eigenvalue of the normalized graph p-Laplacian, see Amghibech [2003] (an analogous result can be shown for the unnormalized graph p-Laplacian).

Theorem 7.19 (Second eigenvalue of graph p-Laplacian). *Let $p > 1$. The global minimum of the functional $Q_p^{(\mathrm{n})}$ is equal to the second eigenvalue λ_p of the graph p-Laplacian $\Delta_p^{(\mathrm{n})}$. The corresponding eigenvector g of $\Delta_p^{(\mathrm{n})}$ is then given as $g = f - \mathrm{mean}_{p,d}(f)\mathbf{1}$ for any global minimizer f of $Q_p^{(\mathrm{n})}$.*

Proof. Let f be a critical point of $\overline{Q}_p^{(\mathrm{n})}$. Note that with $\frac{0}{0} := \infty$, f has to be non-constant. Theorem 3.6 implies that f solves the nonlinear eigenproblem

$$0 \in \partial\Big(\tfrac{1}{2}\sum_{i,j\in V} w_{ij}\,|f_i - f_j|^p\Big) - \lambda\,\partial\Big(\,\|f - \mathrm{mean}_{p,d}(f)\mathbf{1}\|_{p,d}^p\Big)\,,$$

for $\lambda = \overline{Q}_p^{(\mathrm{n})}(f)$. Moreover, in this case this is also a sufficient condition, since for $p > 1$ the denominator is differentiable. The derivative is given as [Amghibech, 2003, Bühler and Hein, 2009b],

$$\tfrac{\partial}{\partial f_k}\Big(\,\|f - \mathrm{mean}_{p,d}(f)\mathbf{1}\|_{p,d}^p\Big) = p\,d_k\,\phi_p\Big(f_k - \mathrm{mean}_{p,d}(f)\Big).$$

Thus with Lemma 7.15 one obtains

$$0 = p\,d_k\,(\Delta_p^{(\mathrm{n})} f)_k - \lambda\, p\,d_k\,\phi_p\Big(f_k - \mathrm{mean}_{p,d}(f)\Big).$$

Since by Theorem 3.6 every eigenvector f with eigenvalue λ satisfies $\lambda = Q_p^{(\mathrm{n})}(f)$, also the reverse direction holds. Using the fact that $\Delta_p^{(\mathrm{n})}$ is invariant under addition of a constant, one can conclude that the vector $g := f - \mathrm{mean}_{p,d}(f)\mathbf{1}$ is a non-constant eigenvector of the graph p-Laplacian according to Def. 7.16. □

Thus one obtains the result that for $p > 1$ the second eigenvector of the normalized and unnormalized graph p-Laplacian is a relaxation of the balanced graph cut criteria RCC_p and NCC_p, respectively. For $p = 2$, we get the well-known fact that the eigenproblem for the second eigenvector of the unnormalized and normalized p-Laplacian $\Delta_2^{(\mathrm{u})}$ and $\Delta_2^{(\mathrm{n})}$ is a relaxation of the ratio cut and the normalized cut.

In the following we relate the optimal values of the relaxed problem to the minimum of the RCC and NCC criterion.

7.3.3 Isoperimetric inequality for p-spectral relaxation

The generalized isoperimetric inequality by Amghibech [2003] establishes a relation between the minimum of the RCC and NCC criterion and the second eigenvalue λ_p of the unnormalized and normalized graph p-Laplacian and therefore generalizes the standard isoperimetric inequality (Cheeger inequality) given for the case $p = 2$. Moreover, one can also use the same proof technique to derive bounds on the RCC and NCC obtained by the p-spectral relaxation. As we will see, for $p \to 1$ these lower and upper bounds will become tighter, which forms the main motivation to use the eigenvectors of the graph p-Laplacian for clustering.

Theorem 7.20 (Generalized isoperimetric inequality). *Denote by λ_p the second eigenvalue of the normalized graph p-Laplacian $\Delta_p^{(\mathrm{n})}$. Then for any $p > 1$,*

$$2^{p-1}\left(\frac{h_{\mathrm{NCC}}}{p}\right)^p \;\leq\; \lambda_p \;\leq\; 2^{p-1}\, h_{\mathrm{NCC}} \;.$$

Denote by λ_p the second eigenvalue of the unnormalized graph p-Laplacian $\Delta_p^{(\mathrm{u})}$. Then for $p > 1$,

$$\left(\frac{2}{\max_{i\in V} d_i}\right)^{p-1}\left(\frac{h_{\mathrm{RCC}}}{p}\right)^p \;\leq\; \lambda_p \;\leq\; 2^{p-1} h_{\mathrm{RCC}} \;.$$

Amghibech [2003] derived the generalized version of Cheeger's inequality for the normalized graph p-Laplacian. In Bühler [2009], Bühler and Hein [2009a], the result was adapted to the unnormalized graph p-Laplacian.

One observes that the bounds on λ_p become tight in the limit $p \to 1$. Furthermore, one can now establish a connection between the optimal Cheeger cut and the cut which is obtained if one performs optimal thresholding of

the second eigenvector of the graph p-Laplacian according to the NCC or RCC criterion. A proof can be found in [Bühler and Hein, 2009b].

Theorem 7.21 (Bounds on obtained cuts). *Let h^*_{RCC} denote the* RCC *obtained by optimal thresholding of the second eigenvector of the unnormalized graph p-Laplacian $\Delta_p^{(\mathrm{u})}$. Then for $p > 1$,*

$$h_{\mathrm{RCC}} \;\leq\; h^*_{\mathrm{RCC}} \;\leq\; p \left(\max_{i \in V} d_i \right)^{\frac{p-1}{p}} \left(h_{\mathrm{RCC}} \right)^{\frac{1}{p}} .$$

Let h^*_{NCC} denote the NCC *obtained by optimal thresholding of the second eigenvector of the normalized graph p-Laplacian $\Delta_p^{(\mathrm{n})}$. Then for $p > 1$,*

$$h_{\mathrm{NCC}} \;\leq\; h^*_{\mathrm{NCC}} \;\leq\; p \left(h_{\mathrm{NCC}} \right)^{\frac{1}{p}} .$$

One observes that the inequalities become tight for $p \to 1$, which implies that the cut found by thresholding converges to the optimal Cheeger cut, which provides the main motivation for p-spectral clustering.

The question remains how to compute the second eigenvector of the unnormalized and normalized graph p-Laplacians $\Delta_p^{(\mathrm{u})}$ and $\Delta_p^{(\mathrm{n})}$. In Bühler and Hein [2009a], a scheme was proposed to compute local minima of $Q_p^{(\mathrm{u})}$ and $Q_p^{(\mathrm{n})}$ based on Newton descent and continuation in p. As in the case of standard spectral clustering, the obtained vector is then transformed back into a partition of the graph via optimal thresholding. Note that it could not be guaranteed that the resulting solutions are in fact globally optimal. As a consequence, one cannot guarantee that the bounds in Theorem 7.21 are achieved by the computed solution. However, in practice a strong improvement in terms of obtained NCC and RCC values is observed compared to the standard spectral relaxation, as we will show in Section 7.7.

In the next section, we will go one step further and consider the case $p = 1$ directly. We will then use our tight relaxation framework from Chapter 4 to show that the optimal Cheeger cut is in fact equal to the second nonlinear eigenvalue of the graph 1-Laplacian, which will lead to an efficient method for the Cheeger cut problem.

7.4 1-Spectral clustering

In the previous sections we have shown that, compared to the standard spectral relaxation using the eigenvectors of the graph Laplacian (see Section 7.2), better guarantees in terms of the obtained Cheeger cut value can be achieved by means of a relaxation based on the functional induced by the graph p-Laplacian (see Section 7.3). The isoperimetric inequality from the last section implies that the bounds on the optimal cut become tight as $p \to 1$ and one converges towards the optimal Cheeger cut.

This suggests to go one step further and consider the case $p = 1$ directly. However, the results from the last section are not directly applicable to the case $p = 1$. The main difficulty arises since in contrast to the case $p > 1$ the involved functionals are non-differentiable. Using the tight relaxation framework introduced in Chapter 4, we can now directly treat the case $p = 1$ and show that the minimum Cheeger cut is equal to the second eigenvalue of the nonlinear graph 1-Laplacian which will be defined in this section. We will then derive an efficient algorithm based on the nonlinear inverse power method discussed in Chapter 5.3.

7.4.1 Tight 1-spectral relaxation of balanced graph cuts

In this section we consider the problem of minimizing the ratio Cheeger cut and normalized Cheeger cut of a graph,

$$\mathrm{RCC}(C, \overline{C}) = \frac{\mathrm{cut}(C, \overline{C})}{\min\{|C|, |\overline{C}|\}} \quad \text{and} \quad \mathrm{NCC}(C, \overline{C}) = \frac{\mathrm{cut}(C, \overline{C})}{\min\{\mathrm{vol}(C), \mathrm{vol}(\overline{C})\}}.$$

We will now derive a tight relaxation of the above balanced graph cut criteria. Consider the functionals

$$Q_1^{(\mathrm{u})}(f) = \frac{R(f)}{S_1^{(\mathrm{u})}(f)} = \frac{\frac{1}{2}\sum_{i,j\in V} w_{ij}\,|f_i - f_j|}{\|f - \mathrm{median}(f)\mathbf{1}\|_1} \quad \text{and} \tag{7.6}$$

$$Q_1^{(\mathrm{n})}(f) = \frac{R(f)}{S_1^{(\mathrm{n})}(f)} = \frac{\frac{1}{2}\sum_{i,j\in V} w_{ij}\,|f_i - f_j|}{\|f - \mathrm{median}_d(f)\mathbf{1}\|_{1,d}},$$

where $\|f\|_{1,d} = \sum_{i\in V} d_i\,|f_i|$ is the weighted 1-norm and the median and its weighted variant are defined as

$$\begin{aligned} \mathrm{median}(f) &= \operatorname*{arg\,min}_{m\in\mathbb{R}} \|f - m\mathbf{1}\|_1 \ \text{ and} \\ \mathrm{median}_d(f) &= \operatorname*{arg\,min}_{m\in\mathbb{R}} \|f - m\mathbf{1}\|_{1,d}\,. \end{aligned} \tag{7.7}$$

We will show that the functionals $Q_1^{(\mathrm{u})}$ and $Q_1^{(\mathrm{n})}$ are extensions of the RCC and NCC criteria. However, in contrast to the extensions used in the spectral and p-spectral relaxation discussed in the previous sections, the relaxation obtained via the functionals $Q_1^{(\mathrm{u})}$ and $Q_1^{(\mathrm{n})}$ is tight, in the sense that the relaxed problem and the original problem are equivalent. This means that the optimal value of the relaxed problem is equal to the optimal value of the original problem, and there exists a simple way to compute minimizer of the original problem from the solution of the continuous problem.

Lemma 7.22. *The set functions* $\min\{|C|, |\overline{C}|\}$ *and* $\min\{\mathrm{vol}(C), \mathrm{vol}(\overline{C})\}$ *are submodular. Moreover, their Lovász extensions are given by the functions* $\|f - \mathrm{median}(f)\mathbf{1}\|_1$ *and* $\|f - \mathrm{median}_d(f)\mathbf{1}\|_{1,d}$*, respectively.*

Proof. We give the proof for the normalized case. The unnormalized case works analogously. Due to the second formulation in Def. 2.12, the Lovász extension is given as

$$R(f) = \sum_{i=1}^{n-1} \left(\widehat{R}(C_i) - \widehat{R}(C_{i+1}) \right) f_i + \widehat{R}(C_n) f_n.$$

Note that $\widehat{R}(C_n) = \min\{d_n, \mathrm{vol}(V) - d_n\}$, which is d_n, if $d_n \leq \frac{1}{2}\mathrm{vol}(V)$, and $\mathrm{vol}(V) - d_n$, else. Moreover, $\forall i = 1, \ldots, n-1$,

$$\widehat{R}(C_i) - \widehat{R}(C_{i+1}) = \begin{cases} d_i, & \text{if } \mathrm{vol}(C_i) \leq \frac{1}{2}\mathrm{vol}(V), \\ -d_i, & \text{if } \mathrm{vol}(C_i) \geq \frac{1}{2}\mathrm{vol}(V) + d_i, \\ \sum_{j=1}^{i-1} d_j - \sum_{j=i+1}^{n} d_j, & \text{if } \mathrm{vol}(C_i) > \frac{1}{2}\mathrm{vol}(V) \\ & \text{and } \mathrm{vol}(C_i) < \frac{1}{2}\mathrm{vol}(V) + d_i. \end{cases}$$

Assume now that $d_n \leq \frac{1}{2}\mathrm{vol}(V)$. Note that the sequence $\mathrm{vol}(C_i)$ is monotonically decreasing for $i = 1, \ldots, n$. Let now $k \in \{1, \ldots, n-1\}$ be the first index such that $\mathrm{vol}(C_k) < \frac{1}{2}\mathrm{vol}(V) + d_k$, i.e. for $i = 1, \ldots, k-1$, the second case in the above expression is applied. For $i = k$ we are now either in the first or third case. Note that one always has $\mathrm{vol}(C_{k+1}) = \mathrm{vol}(C_k) - d_k < \frac{1}{2}\mathrm{vol}(V)$, thus for $i = k+1, \ldots, n-1$, we will be in the first case. We now consider the two cases for $i = k$. Assume first that the third case applies. One can now rewrite the Lovász extension as

$$R(f) = \sum_{i=1}^{k-1} (-d_i) f_i + \left(\sum_{j=1}^{k-1} d_j - \sum_{j=k+1}^{n} d_j \right) f_k + \sum_{i=k+1}^{n} d_i f_i. \tag{7.8}$$

Note that since $\mathrm{vol}(C_{k+1}) < \frac{1}{2}\mathrm{vol}(V)$ and $\mathrm{vol}(\overline{C_k}) < \frac{1}{2}\mathrm{vol}(V)$, it follows that f_k is a weighted median of f. Now assume that for f_k we are in the first case, i.e. $\mathrm{vol}(C_k) \leq \frac{1}{2}\mathrm{vol}(V)$. In this case we obtain

$$R(f) = \sum_{i=1}^{k-1} (-d_i) f_i + \sum_{i=k}^{n} d_i f_i. \tag{7.9}$$

However, due to the fact that $\mathrm{vol}(C_{k-1}) \geq \frac{1}{2}\mathrm{vol}(V) + d_{k-1}$, we must have $\mathrm{vol}(C_k) \geq \frac{1}{2}\mathrm{vol}(V)$, which implies that $\mathrm{vol}(C_k) = \frac{1}{2}\mathrm{vol}(V)$. Moreover, this also implies that $\mathrm{vol}(\overline{C_k}) = \frac{1}{2}\mathrm{vol}(V)$, which implies that every element in the interval (f_{k-1}, f_k) is a median. Denoting by m an element of the interval (f_{k-1}, f_k), one can rewrite (7.9) as

$$R(f) = \sum_{f_i < m} (-d_i) f_i + \left(\sum_{f_i < m} d_i - \sum_{f_i > m} d_i \right) m + \sum_{f_i > m} d_i f_i.$$

Note that the expression in (7.8) can also be rewritten in this form, where here the weighted median is $m = f_k$. Moreover, one easily checks that if $d_n > \frac{1}{2}\operatorname{vol}(V)$ one obtains the same expression. Thus in all cases we obtain

$$R(f) = \sum_{f_i<m} (-d_i)\,(f_i - m) + \sum_{f_i>m} d_i\,(f_i - m) = \sum_{i=1}^{n} d_i\,|f_i - m|\,,$$

which concludes the proof of the second statement. The submodularity follows with Prop. 2.19 from the convexity of the Lovász extensions. □

A direct application of Theorem 4.2 yields the following result, see also Chung [1997], Szlam and Bresson [2010].

Theorem 7.23 (Tight relaxation of RCC and NCC). *It holds that*

$$\begin{aligned} \min_{C\subset V} \operatorname{RCC}(C,\overline{C}) &= \min_{f\in\mathbb{R}^n} Q_1^{(\mathrm{u})}(f) \qquad \textit{and} \\ \min_{C\subset V} \operatorname{NCC}(C,\overline{C}) &= \min_{f\in\mathbb{R}^n} Q_1^{(\mathrm{n})}(f). \end{aligned}$$

Proof. Using the Lovász extensions of numerator and denominator given in Prop. 2.23 and Lemma 7.22, an application of Theorem 4.2 yields

$$\min_{C\subset V} \frac{\operatorname{cut}(C,\overline{C})}{\min\{|C|\,,|\overline{C}|\}} = \min_{f\in\mathbb{R}^n_+} \frac{\frac{1}{2}\sum_{i,j=1}^n w_{ij}\,|f_i - f_j|}{\|f - \operatorname{median}(f)\|_1}$$

The symmetry of the ratio on the right side then yields the final result. □

The tight relaxation of the ratio Cheeger cut was first shown by Szlam and Bresson [2010]. They then proposed a method to minimize the continuous relaxation based on Dinkelbach's method and Bregman iteration [Dinkelbach, 1967, Goldstein and Osher, 2009]. Note that while their method produces comparable cuts to the one we will derive in this section, the convergence can not be guaranteed. Later, Bresson et al. [2012a,b] proposed a modified version of their method for the RCC as well as the tight relaxation of the RCut [Hein and Setzer, 2011], see below. The addition of a proximal term in their inner problem enabled them to prove monotonicity of the sequence as well as convergence of the iterates f^k. Jost et al. [2013] showed that the resulting method is a special case of the RatioDCA-prox from Section 5. Furthermore, recently Bresson et al. [2013] proposed a generalization for partitioning into multiple clusters which will be discussed in Section 7.6.

The above result for the Cheeger cut was extended to a general class of balanced graph cut functions in Hein and Setzer [2011]. In this paper it was shown that tight relaxations exists for the class of problems of the form

$$\min_{C\subset V} \frac{\operatorname{cut}(C,\overline{C})}{\widehat{S}(C)},$$

where $\widehat{S}$ is a symmetric balancing function. In the following we report two different tight relaxations of the normalized cut objective which now follow as special case of Theorem 4.4. Analogously one can proceed for other balanced graph cuts.

Lemma 7.24. *The set function* $\widehat{S}(C) = \frac{\operatorname{vol}(C)\operatorname{vol}(\overline{C})}{\operatorname{vol}(V)}$ *is submodular. An extension of* $\widehat{S}$ *is given by* $S(f) = \frac{1}{2}\|f - \operatorname{mean}_d(f)\|_{1,d}$. *Moreover, the Lovász extension of* $\widehat{S}$ *is given by* $\frac{1}{2}\sum_{i,j\in V}\frac{d_i\, d_j}{\operatorname{vol}(V)}|f_i - f_j|$.

Proof. For the extension $S(f) = \frac{1}{2}\|f - \operatorname{mean}_d(f)\|_{1,d}$ we compute

$$\begin{aligned} S(\mathbf{1}_C) &= \tfrac{1}{2}\sum_{i\in C} d_i\Big|1 - \tfrac{\langle d,\mathbf{1}_C\rangle}{\operatorname{vol}(V)}\Big| + \tfrac{1}{2}\sum_{i\in\overline{C}} d_i\Big|0 - \tfrac{\langle d,\mathbf{1}_C\rangle}{\operatorname{vol}(V)}\Big| \\ &= \tfrac{1}{2}\operatorname{vol}(C)\Big|1 - \tfrac{\operatorname{vol}(C)}{\operatorname{vol}(V)}\Big| + \tfrac{1}{2}\operatorname{vol}(\overline{C})\tfrac{\operatorname{vol}(C)}{\operatorname{vol}(V)} = \tfrac{\operatorname{vol}(C)\operatorname{vol}(\overline{C})}{\operatorname{vol}(V)}, \end{aligned}$$

which shows that S is an extension of $\widehat{S}$. For the statement about the Lovász extension, note that we can write $\widehat{S}$ as

$$\widehat{S}(C) = \frac{(\sum_{i\in C} d_i)\,(\sum_{j\in\overline{C}} d_j)}{\operatorname{vol}(V)} = \sum_{i\in C, j\in\overline{C}} \frac{d_i\, d_j}{\operatorname{vol}(V)}.$$

Thus, $\widehat{S}$ can be interpreted as $\operatorname{cut}(C,\overline{C})$ on a graph with edge weights $w_{ij} = \frac{d_i\, d_j}{\operatorname{vol}(V)}, \forall i,j \in V$. The result then follows from the Lovász extension of the cut function in Prop. 2.23. The submodularity now follows with Prop. 2.19 from the fact that the Lovász extension is convex. □

Theorem 7.25 (Tight relaxation of NCut). *It holds that*

$$\begin{aligned} \min_{C\subset V} \operatorname{NCut}(C,\overline{C}) &= \min_{f\in\mathbb{R}^n} \frac{\sum_{i,j\in V} w_{ij}\,|f_i - f_j|}{\sum_{i,j\in V}\frac{d_i\, d_j}{\operatorname{vol}(V)}|f_i - f_j|} \\ &= \min_{f\in\mathbb{R}^n} \frac{\sum_{i,j\in V} w_{ij}\,|f_i - f_j|}{\|f - \operatorname{mean}_d(f)\|_{1,d}}. \end{aligned}$$

Proof. Using the Lovász extensions of numerator and denominator given in Prop. 2.23 and Lemma 7.24, we directly obtain the first result via Theorem 4.2, using the fact that the above Lovász extensions are symmetric. For the second statement, noting that the extension given in Lemma 7.24 is convex, 1-homogeneous and non-negative, we can then apply Theorem 4.4, which yields the result. □

The above theorem gives two different tight relaxations of the NCut criterion by choosing two different extensions of the balancing function $\widehat{S}(C)$. The function $\frac{1}{2}\|f - \text{mean}_d(f)\|_{1,d}$ was used in Hein and Setzer [2011]. As observed by Jost et al. [2013], the fact that the Lovász extension is maximal in the class of 1-homogeneous extensions suggests that the Lovász extension should lead to better performance. Indeed, it was experimentally confirmed in Jost et al. [2013] on several graphs that the Lovász extension consistently leads to better results in terms of the obtained objective value.

We will discuss the application of the nonlinear inverse power method to the above functionals in the Section 7.4.3. In the following, we will show the relation to a nonlinear eigenproblem involving the graph 1-Laplacian, which will be introduced below.

7.4.2 Connection to eigenvectors of the graph 1-Laplacian

In this section we will show the relation of the functionals introduced in the last section to a nonlinear eigenproblem of the form $0 \in \partial R(f) - \lambda \partial S(f)$. In contrast to the spectral and p-spectral case in the previous sections, the numerator and denominator in the functionals $Q_1^{(\mathrm{u})}$ and $Q_1^{(\mathrm{n})}$ in (7.6) are non-differentiable, thus we will obtain set-valued operators in this section. We first give the definitions of unnormalized and normalized graph 1-Laplacian.

Definition 7.26 (Graph 1-Laplacian). *Let $i \in V$, then the unnormalized graph 1-Laplacian $\Delta_1^{(\mathrm{u})}$ and normalized graph 1-Laplacian $\Delta_1^{(\mathrm{n})}$ are defined as*

$$(\Delta_1^{(\mathrm{u})} f)_i = \left\{ \sum_{j=1}^n w_{ij} u_{ij} \,|\, u_{ij} = -u_{ji}, u_{ij} \in \text{sign}(f_i - f_j) \right\}$$

$$(\Delta_1^{(\mathrm{n})} f)_i = \left\{ \tfrac{1}{d_i} \sum_{j=1}^n w_{ij} u_{ij} \,|\, u_{ij} = -u_{ji}, u_{ij} \in \text{sign}(f_i - f_j) \right\},$$

where $\text{sign}(x) = \begin{cases} -1, & x < 0, \\ [-1,1], & x = 0, \\ 1, & x > 0. \end{cases}$

The origin of the above definition will become clear in the next lemma, where we relate the graph 1-Laplacian to the functional $\frac{1}{2}\sum_{i,j=1}^n w_{ij}\,|f_i - f_j|$.

Lemma 7.27. *The following statements hold.*

$$\tfrac{1}{2} \sum_{i,j=1}^n w_{ij}\,|f_i - f_j| \;=\; \langle f, \Delta_1^{(\mathrm{u})} f \rangle \;=\; \langle f, \Delta_1^{(\mathrm{n})} f \rangle_d,$$

$$\partial\Big(\tfrac{1}{2} \sum_{i,j=1}^n w_{ij}\,|f_i - f_j| \Big) \;=\; \Delta_1^{(\mathrm{u})} f \;=\; D\big(\Delta_1^{(\mathrm{n})} f\big).$$

Proof. We have for each $k = 1, \ldots, n$,

$$\partial\Big(\tfrac{1}{2}\sum_{i,j=1}^{n} w_{ij}\,|f_i - f_j|\Big)_k = \tfrac{1}{2}\sum_{i,j=1}^{n} w_{ij}\;\partial_k\left(|f_i - f_j|\right).$$

Note that for $x \in \mathbb{R}$, the subdifferential of $|x|$ is given as

$$\operatorname{sign}(x) = \begin{cases} -1, & x < 0, \\ [-1,1], & x = 0, \\ 1, & x > 0. \end{cases}$$

As each of the terms $f_i - f_j$ can be obtained via an affine transformation of f, we can apply the chain rule for subdifferentials (see e.g. Theorem 23.9 in Rockafellar [1970]). The subdifferential is then given as

$$\left\{\tfrac{1}{2}\sum_{i,j=1}^{n} w_{ij}\,v_{ij}\;(\delta_{i=k} - \delta_{j=k}) \;\mid v_{ij} \in \operatorname{sign}(f_i - f_j),\ \forall i,j \in V\right\},$$

where we used the notation $\delta_A = 1$ if A holds, and 0 else. Using the symmetry of W, one can rewrite this as

$$\left\{\tfrac{1}{2}\sum_{j=1}^{n} w_{kj}\;(v_{kj} - v_{jk}) \;\mid v_{ij} \in \operatorname{sign}(f_i - f_j),\ \forall i,j \in V\right\}.$$

Note that while $v_{ij} = -v_{ji}$ if $f_i \neq f_j$, this is in general not the case if $f_i = f_j$, thus the anti-symmetry does not hold for the v_{ij} in general. We now apply the substitution $u_{ij} = \frac{1}{2}(v_{ij} - v_{ji})$, which implies that $u_{ij} \in \operatorname{sign}(f_i - f_j)$ as well as $u_{ij} = -u_{ji}$, $\forall i,j \in V$. Thus we can rewrite the subdifferential as

$$\left\{\sum_{j=1}^{n} w_{kj}u_{kj} \,|\, u_{ij} = -u_{ji}, u_{ij} \in \operatorname{sign}(f_i - f_j),\ \forall i,j \in V\right\},$$

which is just the unnormalized 1-Laplacian from Def. 7.26. The equality $\langle f, \Delta_1^{(\mathrm{u})} f\rangle = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\,|f_i - f_j|$ now follows from Lemma 2.8 and the fact that the function is 1-homogeneous. The statements for $\Delta_1^{(\mathrm{n})}$ can be shown by noting that $\Delta_1^{(\mathrm{u})} f = D(\Delta_1^{(\mathrm{n})} f)$ for all $f \in \mathbb{R}^n$. □

Similarly to the case $p > 1$, the definition of eigenvectors of the graph 1-Laplacian can be motivated via the critical points of a certain nonlinear Rayleigh quotient. Consider now the following functionals associated to the graph 1-Laplacians $\Delta_1^{(\mathrm{u})}$ and $\Delta_1^{(\mathrm{n})}$,

$$\overline{Q}_1^{(\mathrm{u})}(f) = \frac{\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}|f_i - f_j|}{\|f\|_1} \quad \text{and} \quad \overline{Q}_1^{(\mathrm{n})}(f) = \frac{\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}|f_i - f_j|}{\|f\|_{1,d}}.$$

The following proposition is an application of Theorem 3.6 in Chapter 3.

Proposition 7.28. *A necessary condition for $f \in \mathbb{R}^n$ being a critical point of $\overline{Q}_1^{(\mathrm{n})}$ is given as $0 \in \Delta_1^{(\mathrm{n})} f - \lambda \operatorname{sign}(f)$, where $\lambda = \overline{Q}_1^{(\mathrm{n})}(f)$. Moreover, if the above condition is fulfilled for some $\lambda \in \mathbb{R}$ and $f \in \mathbb{R}^n$, then $\lambda = \overline{Q}_1^{(\mathrm{n})}(f)$. The analogous result holds for the unnormalized 1-Laplacian.*

Proof. Note that one has $\partial \left(\|f\|_1 \right)_k = \operatorname{sign}(f_k),\ \forall k = 1, \ldots, n$, and analogously one obtains $\partial \left(\|f\|_{1,d} \right)_k = d_k \operatorname{sign}(f_k),\ \forall k = 1, \ldots, n$. The result then follows directly from Theorem 3.6 and Lemma 7.27. □

This result forms the motivation for the following definition of eigenvectors and eigenvalues of the 1-Laplacian.

Definition 7.29 (Eigenvalues of graph 1-Laplacian). *The real number λ is called an eigenvalue for the normalized 1-Laplacian $\Delta_1^{(\mathrm{n})}$ if there exists a vector $f \in \mathbb{R}^n$ such that*

$$0 \in (\Delta_1^{(\mathrm{n})} f)_i - \lambda \operatorname{sign}(f)_i \quad \forall\, i = 1, ..., n.$$

The vector f is called an eigenvector of $\Delta_1^{(\mathrm{n})}$.

Similarly, one defines the eigenvectors and eigenvalues of $\Delta_1^{(\mathrm{u})}$. One can now give the following result for the smallest eigenvalue.

Lemma 7.30. *The eigenvector of $\Delta_1^{(\mathrm{u})}$ and $\Delta_1^{(\mathrm{n})}$ corresponding to the smallest eigenvalue is the constant vector, with eigenvalue 0.*

Proof. One can easily check that the nonlinear eigenproblem is fulfilled in both cases by setting $f = \mathbf{1}$ and $\lambda = 0$. Moreover, by Theorem 7.28 any eigenvalue λ and eigenvector f must satisfy $\lambda = \overline{Q}_1^{(\mathrm{n})}(f)$. By the non-negativity of $\overline{Q}_1^{(\mathrm{n})}$ it follows that $\lambda = 0$ has to be the smallest eigenvalue. Analogously one proceeds for the unnormalized case. □

The following is a characterization of the non-constant eigenvectors of the unnormalized and normalized graph 1-Laplacians $\Delta_1^{(\mathrm{u})}$ and $\Delta_1^{(\mathrm{n})}$.

Lemma 7.31. *For any non-constant eigenvector f of $\Delta_1^{(\mathrm{u})}$ it holds that $\operatorname{median}(f) = 0$, and for any non-constant eigenvector f of $\Delta_1^{(\mathrm{n})}$ it holds that $\operatorname{median}_d(f) = 0$. The corresponding eigenvalues satisfy $\lambda > 0$.*

Proof. By Theorem 7.28 any eigenvalue λ and eigenvector f must satisfy $\lambda = \overline{Q}_1^{(\mathrm{n})}(f)$. Using the assumption that the graph is connected, the functional $\overline{Q}_1^{(\mathrm{n})}$ can only be zero if f is constant. Thus any non-constant eigenvector f must have $\lambda > 0$. Let f be an eigenvector of the normalized graph 1-Laplacian with eigenvalue $\lambda > 0$. Then $\forall i, j \in V$ there must exist

u_{ij} with $u_{ij} = -u_{ji}$ and $u_{ij} \in \operatorname{sign}(f_i - f_j)$, as well as α_i with $\alpha_i \in \operatorname{sign}(f_i)$ such that

$$0 = \tfrac{1}{d_i} \sum_{j=1}^{n} w_{ij} u_{ij} - \lambda \alpha_i.$$

Multiplying by d_i and then summing over i yields

$$\lambda \sum_{i \in V} \alpha_i d_i = \sum_{i,j \in V} w_{ij} u_{ij} = \sum_{i>j} w_{ij} u_{ij} + \sum_{i<j} w_{ij} u_{ij} + \sum_{i=j} w_{ij} u_{ij} = 0,$$

where we used the anti-symmetry of u_{ij} as well as the fact that W is symmetric. As $\lambda > 0$ this implies $\sum_{i \in V} \alpha_i d_i = 0$. Thus one obtains

$$0 = \sum_{i \in V} \alpha_i d_i = \sum_{i \in C_+} d_i - \sum_{i \in C_-} d_i + \sum_{i \in C_0} \alpha_i d_i,$$

where $C_+ = \{i \in V \mid f_i > 0\}$, and C_- and C_0 are defined analogously. Thus we obtain $-\operatorname{vol}(C_0) \leq \operatorname{vol}(C_+) - \operatorname{vol}(C_-) \leq \operatorname{vol}(C_0)$, which implies with $\operatorname{vol}(C_+) + \operatorname{vol}(C_-) + \operatorname{vol}(C_0) = \operatorname{vol}(V)$ that $\operatorname{vol}(C_+) \leq \frac{1}{2}\operatorname{vol}(V)$ and $\operatorname{vol}(C_-) \leq \frac{1}{2}\operatorname{vol}(V)$. Let now f_+^* be the smallest non-negative value of f, i.e. $f_+^* = \arg\min\{f_i \mid f_i \geq 0\}$, and one similarly defines the largest non-positive value f_-^*. Then $\forall f^* \in \left[f_-^*, f_+^*\right]$ it holds that $\operatorname{vol}(\{f_i \mid f_i > f^*\}) \leq \operatorname{vol}(C_+) \leq \frac{1}{2}\operatorname{vol}(V)$, and $\operatorname{vol}(\{f_i \mid f_i < f^*\}) \leq \operatorname{vol}(C_-) \leq \frac{1}{2}\operatorname{vol}(V)$. Thus every element in the set $\left[f_-^*, f_+^*\right]$, which contains zero, is a weighted median. If f contains the value $f_k = 0$, then this is the unique weighted median. The result for the unnormalized case works analogously. □

Due to the set-valued nature of the eigenproblem for the graph 1-Laplacian as well as the fact that in contrast to the case $p > 1$ it constitutes only a necessary condition for the critical points of the nonlinear Rayleigh quotients $Q_1^{(\mathrm{u})}$ and $Q_1^{(\mathrm{n})}$, the technique used in the case $p > 1$ is not sufficient to show that the global minima of $Q_1^{(\mathrm{u})}$ and $Q_1^{(\mathrm{n})}$ are equal to the second eigenvalues of the graph 1-Laplacians $\Delta_1^{(\mathrm{u})}$ and $\Delta_1^{(\mathrm{n})}$. However, this fact can be proven using the techniques applied in the convergence proof of the RatioDCA and nonlinear IPM in Chapter 5. A variant of the following lemma was used in Chapter 5 to show that the algorithms RatioDCA and nonlinear IPM create a decreasing sequence of objective values and converge to the solution of a nonlinear eigenproblem.

Lemma 7.32. *Define for any $g \in \mathbb{R}^n$, $\Phi_g(f) := \frac{1}{2}\sum_{i,j=1}^n w_{ij}|f_i - f_j| - Q_1^{(\mathrm{n})}(g)\,\langle f, s(g)\rangle$, where $s(g) \in \partial S_1^{(\mathrm{n})}(g)$. Assume there exists an $f \in \mathbb{R}^n$ with $\|f\|_2 \leq 1$ such that $\Phi_g(f) < 0$. Then $Q_1^{(\mathrm{n})}(f) < Q_1^{(\mathrm{n})}(g)$.*

Proof. The Lemma is an application of Lemma 5.5, choosing $R(f) = \frac{1}{2}\sum_{i,j=1}^n w_{ij}|f_i - f_j|$ and $S(f) = \|f - \operatorname{median}_d(f)\mathbf{1}\|_{1,d}$. □

An analogous statement can be made in the unnormalized case. In the following we will give an explicit characterization of an element of the subdifferential of $S_1^{(\mathrm{n})}$ and $S_1^{(\mathrm{u})}$. We use the notation $\mathrm{vol}(g_+) = \sum_{g_i>0} d_i$ and $|g_+| = |\{g_i \mid g_i > 0\}|$, and analogously define $\mathrm{vol}(g_-)$, $|g_-|$, $\mathrm{vol}(g_0)$ and $|g_0|$.

Lemma 7.33. *An element* $s(f) \in \partial\big(\|f - \mathrm{median}_d(f)\mathbf{1}\|_{1,d}\big)$ *is given by*

$$s(f)_i = \begin{cases} d_i \operatorname{sign}(g_i), & \text{if } g_i \neq 0, \\ -d_i \frac{\mathrm{vol}(g_+) - \mathrm{vol}(g_-)}{\mathrm{vol}(g_0)}, & \text{if } g_i = 0, \end{cases}$$

where $g = f - \mathrm{median}_d(f)$. *Moreover, one has* $s(f) \in \partial\big(\|g\|_{1,d}\big)$. *An element* $s^{(\mathrm{u})}(f) \in \partial\big(\|f - \mathrm{median}(f)\mathbf{1}\|_1\big)$ *is given by*

$$s^{(\mathrm{u})}(f)_i = \begin{cases} \operatorname{sign}(g_i), & \text{if } g_i \neq 0, \\ -\frac{|g_+| - |g_-|}{|g_0|}, & \text{if } g_i = 0, \end{cases}$$

where $g = f - \mathrm{median}(f)$. *Moreover, one has* $s^{(\mathrm{u})}(f) \in \partial\big(\|f\|_1\big)$.

Proof. Note that since the vector g has weighted median 0, it holds that $|\mathrm{vol}(g_+) - \mathrm{vol}(g_-)| \leq \mathrm{vol}(g_0)$, and therefore $|s(f)_i| \leq d_i, \forall i \in V$. Thus $s(f)_i \in d_i \operatorname{sign}(g_i) = \partial\big(\|g\|_{1,d}\big)_i$ which proves the second statement for the normalized case. For the first statement, let $h \in \mathbb{R}^n$. Then one has

$$\begin{aligned} \langle s(f), h - f\rangle =& \sum_{i\in V} s(f)_i \,(h_i - \mathrm{median}_d(h)) - \sum_{i\in V} s(f)_i \,(f_i - \mathrm{median}_d(f)) \\ &+ \sum_{i\in V} s(f)_i \,(\mathrm{median}_d(h) - \mathrm{median}_d(f)) \\ \leq& \sum_{i\in V} d_i \,|h_i - \mathrm{median}_d(h)| - \sum_{i\in V} d_i \,|f_i - \mathrm{median}_d(f)|\,, \end{aligned}$$

where we have used that $\sum_{i\in V} s(f)_i = 0$. Thus $S_1^{(\mathrm{n})}(f) + \langle s(f), h - f\rangle \leq S_1^{(\mathrm{n})}(h)$, which completes the proofs for the normalized case. The proofs for the unnormalized case work analogously. □

Finally, the following theorem relates the solution of the tight relaxation of the NCC and RCC to the second eigenvalue of the graph 1-Laplacian.

Theorem 7.34 (Second eigenvalue of graph 1-Laplacian). *The global minimum of the functional* $Q_1^{(\mathrm{n})}$ *is equal to the second eigenvalue* λ *of the graph* 1*-Laplacian* $\Delta_1^{(\mathrm{n})}$. *The corresponding eigenvector* g *of* $\Delta_1^{(\mathrm{n})}$ *is given as* $g = f - \mathrm{median}_d(f)\mathbf{1}$ *for any global minimizer* f *of* $Q_1^{(\mathrm{n})}$. *Moreover, let* λ *be the second eigenvalue of* $\Delta_1^{(\mathrm{n})}$, *then if* G *is connected it holds* $\lambda = h_{\mathrm{NCC}}$. *The analogous statement holds for the unnormalized graph 1-Laplacian.*

Proof. Note that we have by Theorem 7.23,

$$\min_{f\in\mathbb{R}^n} Q_1^{(\mathrm{n})}(f) = \min_{C\subset V} \mathrm{NCC}(C,\overline{C}) = Q_1^{(\mathrm{n})}(f^*),$$

where $f^* = \mathbf{1}_{C^*}$ and C^* is the set achieving the optimal NCC. We first prove that $f^* - \mathrm{median}_d(f)\mathbf{1}$ is an eigenvector of the normalized graph 1-Laplacian according to Def. 7.29. To show this, consider the functional Φ_{f^*} introduced in Lemma 7.32,

$$\Phi_{f^*}(f) := \tfrac{1}{2}\sum_{i,j=1}^{n} w_{ij}|f_i - f_j| - Q_1^{(\mathrm{n})}(f^*)\,\langle f, s(f^*)\rangle\,,$$

where $s(f^*)$ is given as in Lemma 7.33. Assume for the sake of contradiction that there exists a vector $f^{**}\in\mathbb{R}^n$ with $\|f^{**}\|_2 \le 1$ such that $\Phi_{f^*}(f^{**}) < 0$. Then by Lemma 7.32, this implies that $Q_1^{(\mathrm{n})}(f^{**}) < Q_1^{(\mathrm{n})}(f^*)$. By Lemma 4.1, optimal thresholding of the vector f^{**} will lead to a set C' such that $Q_1^{(\mathrm{n})}(f^{**}) \ge Q_1^{(\mathrm{n})}(\mathbf{1}_{C'})$. Thus we must have $\mathrm{NCC}(C^*,\overline{C}^*) > \mathrm{NCC}(C',\overline{C}')$, which is a contradiction to the fact that C^* is optimal.

Thus our assumption must be wrong and the function $\Phi_{f^*}(f)$ is non-negative in the unit ball. In fact, using the 1-homogeneity of Φ_{f^*}, one can conclude that the function $\Phi_{f^*}(f)$ is non-negative on $\mathbb{R}^n$. Therefore, since $\Phi_{f^*}(f^*) = 0$, the vector f^* is a global minimizer of Φ_{f^*}, which implies that

$$0 \in \partial\Phi_{f^*}(f^*) = \Delta_1^{(\mathrm{u})}(f^*) - Q_1^{(\mathrm{n})}(f^*)s(f^*). \tag{7.10}$$

The first statement in Lemma 7.33 implies that f^* is an eigenvector associated to the eigenproblem

$$0 \in \partial\Big(\tfrac{1}{2}\sum_{i,j=1}^{n} w_{ij}|f_i - f_j|\Big) - \lambda\,\partial\Big(\,\|f - \mathrm{median}_d(f)\mathbf{1}\|_{1,d}\,\Big), \tag{7.11}$$

with eigenvalue $\lambda^* = Q_1^{(\mathrm{n})}(f^*) = h_{\mathrm{NCC}} > 0$ (assuming that the graph is connected). Unfortunately this is not useful since it is the wrong eigenproblem. However, we can use the second statement in Lemma 7.33 to conclude from (7.10) that $g^* := f^* - \mathrm{median}_d(f^*)\mathbf{1}$ is also an eigenvector with the same eigenvalue λ^* associated to the eigenproblem

$$0 \in \partial\Big(\tfrac{1}{2}\sum_{i,j=1}^{n} w_{ij}|f_i - f_j|\Big) - \lambda\,\partial\Big(\,\|f\|_{1,d}\,\Big), \tag{7.12}$$

which is the eigenproblem from Def. 7.29. Note that while the statement in (7.11) could have been derived directly using Theorem 3.6, the explicit subgradient $s(f^*)$ was needed to show that (7.12) holds.

Thus we have proven that g^* is an eigenvector of the graph 1-Laplacian with eigenvalue λ^*. What is left to show is that λ^* is equal to the smallest eigenvalue λ_2 of the normalized graph 1-Laplacian. To see this, first observe that the eigenvector f_2 corresponding to λ_2 is non-constant, and therefore has weighted median zero by Lemma 7.31. Since $\lambda^* > 0$, we must have $\lambda^* \geq \lambda_2$. On the other hand, we have

$$\lambda_2 = \overline{Q}_1^{(\mathrm{n})}(f_2) \geq \min_{\substack{f \in \mathbb{R}^n \\ \text{s.t. } \mathrm{median}_d(f)=0}} \overline{Q}_1^{(\mathrm{n})}(f) = \min_{f \in \mathbb{R}^n} Q_1^{(\mathrm{n})}(f) = \lambda^*,$$

which shows that $\lambda^* = \lambda_2$. The unnormalized case works analogously. □

Thus the NCC and RCC problems can be solved globally optimal by computing the second eigenvalue of the corresponding graph 1-Laplacian. In the next section we will apply the RatioDCA/nonlinear IPM to the ratios $Q_1^{(\mathrm{n})}$ and $Q_1^{(\mathrm{u})}$ associated to normalized or unnormalized graph 1-Laplacian.

7.4.3 Solution via nonlinear inverse power method

In this section we apply the RatioDCA from Section 5.4 to the tight relaxation of the RCC and NCC criteria in Theorem 7.23, given as

$$\min_{f \in \mathbb{R}^n} \frac{\frac{1}{2}\sum_{i,j \in V} w_{ij} \left| f_i - f_j \right|}{\| f - \mathrm{median}(f)\mathbf{1} \|_1} \quad \text{and} \quad \min_{f \in \mathbb{R}^n} \frac{\frac{1}{2}\sum_{i,j \in V} w_{ij} \left| f_i - f_j \right|}{\| f - \mathrm{median}_d(f)\mathbf{1} \|_{1,d}}.$$

As numerator and denominator are convex, the algorithm boils down to the nonlinear IPM from Section 5.3. Recall that at each step k of the algorithm, we need to solve an inner problem of the form

$$f^{k+1} = \operatorname*{arg\,min}_{\|u\|_2 \leq 1} \left\{ R(u) - \lambda^k \langle u, s(f^k) \rangle \right\}, \qquad \text{where } s(f^k) \in \partial S(f^k).$$

For the RCC criterion, we have $R(f) = \frac{1}{2}\sum_{i,j \in V} w_{ij} \left| f_i - f_j \right|$ and $S(f) = \| f - \mathrm{median}(f)\mathbf{1} \|_1$. An element of the subdifferential of S has been derived in Lemma 7.33. Applying the nonlinear IPM then leads to the algorithmic scheme in Alg. 14.

A slightly different derivation of the algorithm was given in Hein and Bühler [2010]. Instead of directly applying the nonlinear IPM to the ratio $Q_1^{(\mathrm{u})}$, first the eigenproblem associated with the ratio $\overline{Q}_1^{(\mathrm{u})}$ was considered, given as $0 \in \Delta_1 f - \lambda \, \mathrm{sign}(f)$. Applying the nonlinear IPM to the ratio $\overline{Q}_1^{(\mathrm{u})}$ leads to an algorithm converging to the smallest eigenvector of the graph 1-Laplacian, which is the constant vector. Thus a modification to the algorithm was proposed to achieve convergence to a nonconstant eigenvector associated to the above eigenproblem, which again leads to Alg. 14.

Furthermore, note that in Hein and Bühler [2010], the analysis of convergence was done with respect to the sequence g^k. Moreover, the update

14 Computing a nonconstant 1-eigenvector of the unnormalized graph 1-Laplacian $\Delta_1^{(\mathrm{u})}$

Input: weight matrix W
Initialization: nonconstant f^0 with $\left\|f^0\right\|_2 \leq 1$, accuracy ϵ
repeat
  $g^k = f^k - \operatorname{median}\left(f^k\right)\mathbf{1}$
  $s_i^k = \begin{cases} \operatorname{sign}(g_i^k), & \text{if } g_i^k \neq 0, \\ -\frac{|g_+^k| - |g_-^k|}{|g_0^k|}, & \text{if } g_i^k = 0 \end{cases}$
  $f^{k+1} = \underset{\|f\|_2^2 \leq 1}{\arg\min} \left\{ \frac{1}{2} \sum_{i,j=1}^n w_{ij}|f_i - f_j| - \lambda^k \left\langle f, s^k \right\rangle \right\}$
  $\lambda^{k+1} = Q_1^{(\mathrm{u})}(f^{k+1})$
until $\frac{\left|\lambda^{k+1} - \lambda^k\right|}{\lambda^k} < \epsilon$

of the λ^k was computed as $\lambda^{k+1} = \overline{Q}_1^{(\mathrm{u})}(g^{k+1})$. The equivalence between the two algorithms follows with the fact that $\overline{Q}_1^{(\mathrm{u})}(g^{k+1}) = Q_1^{(\mathrm{u})}(f^{k+1})$.

The following properties are corollaries of Prop. 5.9, Theorem 5.10 and Theorem 5.13 from Section 5.4.

Proposition 7.35. *The sequences f^k and g^k produced by Alg. 14 satisfy $Q_1^{(\mathrm{u})}(f^k) > Q_1^{(\mathrm{u})}(f^{k+1})$ and $\overline{Q}_1^{(\mathrm{u})}(g^k) > \overline{Q}_1^{(\mathrm{u})}(g^{k+1})$ for all $k \geq 0$ or the sequence terminates.*

Proof. The statement regarding $Q_1^{(\mathrm{u})}$ and the sequences f^k is a direct corollary of Prop. 5.9. The second statement follows with $\overline{Q}_1^{(\mathrm{u})}(g^k) = \overline{Q}_1^{(\mathrm{u})}(f^k - \operatorname{median}(f^k)\mathbf{1}) = Q_1^{(\mathrm{u})}(f^k)$. □

Theorem 7.36 (Convergence). *The sequence g^k produced by Alg. 14 has a convergent subsequence that converges to an eigenvector f^* of the unnormalized graph 1-Laplacian. The corresponding eigenvalue is given as $\lambda^* = \lim_{k\to\infty} Q(f^k) \in \left[h_{\mathrm{RCC}}, Q_1^{(\mathrm{u})}(f^0)\right]$.*

Proof. By Theorem 5.10, the sequence f^k converges to an eigenvector f^* of the eigenproblem associated with the functional $Q_1^{(\mathrm{u})}$, given as

$$0 \in \partial\Big(\tfrac{1}{2} \sum_{i,j\in V} w_{ij}\,|f_i - f_j|\,\Big) - \lambda\,\partial\Big(\,\|f - \operatorname{median}(f)\mathbf{1}\|_1\,\Big),$$

with eigenvalue $\lambda^* = Q_1^{(\mathrm{u})}(f^*)$. More precisely, by the proof of Theorem 5.10, we will have

$$0 \in \Delta_1^{(\mathrm{u})}(f^*)_i - \lambda^*\, s(f^*)_i,$$

where for $f \in \mathbb{R}^n$, $s(f)$ is the specific element of the subdifferential which is used in the algorithm (see Lemma 7.33). By Lemma 7.33, it holds that $s(f^*) \in \partial \left(\|f\|_1 \right) (f^* - \text{median}(f^*)\mathbf{1}) = \text{sign}(f^* - \text{median}(f^*)\mathbf{1})$, which shows that

$$0 \in \Delta_1^{(\text{u})}(g^*)_i - \lambda^* \text{ sign}(g_i^*),$$

where $g^* = f^* - \text{median}(f^*)\mathbf{1}$. Thus the sequence $g^k = f^k - \text{median}(f^k)\mathbf{1}$ converges to an eigenvector of the graph 1-Laplacian with eigenvalue λ^*.

Note that for any constant vector, the inner problem has objective value zero. Thus the minimizer of the inner problem is either non-constant, or the sequence f^k terminates, in which case the previous non-constant f^k is also a minimizer. Thus we can conclude that the sequence f^k is non-constant. This implies that also the sequence g^k is non-constant. Thus the eigenvector g^* is non-constant and by Lemma 7.31 we must have $\lambda^* \geq \lambda_2 > 0$, where λ_2 is the second smallest eigenvalue of the graph 1-Laplacian. By Theorem 7.34 we have $\lambda_2 = h_{\text{RCC}}$, which concludes the proof. □

Note that the above theorem only gives a guarantee that one obtains a non-constant eigenvector of the graph 1-Laplacian, not necessarily the second one. However, it is clear that there cannot exist a polynomial time algorithm which can guarantee to solve the problem exactly, since the original (equivalent) combinatorial problem is NP hard. Thus in practice one runs the algorithm several times with random initializations and takes the result achieving the best objective value.

Moreover, given a partition of the graph, Theorem 5.13 implies that performing one run initialized with the corresponding indicator function either directly terminates or finds a better partition. Thus in particular it makes sense to always perform one run initialized with the solution of the standard spectral relaxation. As we will see in the experiments, this strategy will lead to a performance of our method which is superior to standard spectral clustering by a large margin.

Theorem 7.37 (Cut improvement). *Let C be any set, f denote the result of Alg. 14 after initializing with the vector $\frac{1}{|C|}\mathbf{1}_C$, and C_f be the set obtained by optimal thresholding of f. Either Alg. 14 terminates after one iteration, or it holds that* $\text{RCC}(C, \overline{C}) > \text{RCC}(C_f, \overline{C_f})$.

Proof. This is a direct corollary of Theorem 5.13. □

For the normalized case, the derivation is analogous and leads to the method in Alg. 15. The analogous statements to Prop. 7.35, Theorem 7.36 and Theorem 7.37 can also be made for Alg. 15.

15 Computing a nonconstant 1-eigenvector of the normalized graph 1-Laplacian $\Delta_1^{(\mathrm{n})}$

Input: weight matrix W
Initialization: nonconstant f^0 with $\left\|f^0\right\|_2 \leq 1$, accuracy ε,
repeat
  $g^k = f^k - \mathrm{median}_d\left(f^k\right)\mathbf{1}$
  $s_i^k = \begin{cases} d_i \operatorname{sign}(g_i^k), & \text{if } g_i^k \neq 0, \\ d_i \frac{\mathrm{vol}(g_-^k) - \mathrm{vol}(g_+^k)}{\mathrm{vol}(g_0^k)}, & \text{if } g_i^k = 0 \end{cases}$
  $f^{k+1} = \operatorname*{arg\,min}_{\|f\|_2^2 \leq 1} \left\{ \frac{1}{2} \sum_{i,j=1}^n w_{ij}|f_i - f_j| - \lambda^k \left\langle f, s^k \right\rangle \right\}$
  $\lambda^{k+1} = Q_1^{(\mathrm{n})}(f^{k+1})$
until $\frac{\left|\lambda^{k+1} - \lambda^k\right|}{\lambda^k} < \epsilon$

7.4.4 Solution of the inner problem

The inner problem is convex, thus a solution can be computed by any standard method for solving convex nonsmooth programs, see the discussion in Chapter 6. However, in this particular case we can exploit the structure of the problem and use the equivalent dual formulation of the inner problem.

Lemma 7.38. *The inner problem is equivalent to*

$$\min_{\substack{\alpha \in \mathbb{R}^{|E|} \\ \|\alpha\|_\infty \leq 1}} \Psi(\alpha) := \left\| A\alpha - \lambda^k s^k \right\|_2^2,$$

where the operator $A : \mathbb{R}^{|E|} \to \mathbb{R}$ *is for* $\alpha \in \mathbb{R}^{|E|}$ *defined as* $(A\alpha)_i := \frac{1}{2} \sum_{j \,|\, (i,j) \in E} w_{ij} \left(\alpha_{ij} - \alpha_{ji}\right)$. *The gradient of* Ψ *is given as*

$$(\nabla \Psi(\alpha))_{rs} = w_{rs} \left(z_r - z_s\right), \qquad \text{where } z = A\alpha - \lambda^k s^k.$$

Moreover, an upper bound on the Lipschitz constant of the gradient of Ψ *is given by* $2 \max_r \sum_{s \,|\, (r,s) \in E} w_{rs}^2$.

Proof. First, we note that

$$\begin{aligned} \tfrac{1}{2} \sum_{i,j=1}^n w_{ij}|u_i - u_j| &= \max_{\substack{\alpha \in \mathbb{R}^{|E|} \\ \|\alpha\|_\infty \leq 1}} \tfrac{1}{2} \sum_{(i,j) \in E} w_{ij}(u_i - u_j)\alpha_{ij} \\ = \max_{\substack{\alpha \in \mathbb{R}^{|E|} \\ \|\alpha\|_\infty \leq 1}} \tfrac{1}{2} \sum_{(i,j) \in E} w_{ij}(\alpha_{ij} - \alpha_{ji})u_i &= \max_{\substack{\alpha \in \mathbb{R}^{|E|} \\ \|\alpha\|_\infty \leq 1}} \left\langle u, A\alpha \right\rangle \ . \end{aligned}$$

Both u and α are constrained to lie in non-empty compact, convex sets, and thus we can reformulate the inner objective by the standard min-max-

theorem (see e.g. Corollary 37.3.2. in Rockafellar [1970]) as follows:

$$\begin{aligned}&\min_{\|u\|_2\leq 1}\ \max_{\substack{\alpha\in\mathbb{R}^{|E|}\\ \|\alpha\|_\infty\leq 1}}\ \big\langle u, A\alpha\big\rangle - \lambda^k\big\langle u, s^k\big\rangle\\ =\ &\max_{\substack{\alpha\in\mathbb{R}^{|E|}\\ \|\alpha\|_\infty\leq 1}}\ \min_{\|u\|_2\leq 1}\big\langle u, A\alpha-\lambda^k s^k\big\rangle = \max_{\substack{\alpha\in\mathbb{R}^{|E|}\\ \|\alpha\|_\infty\leq 1}}\ -\big\|A\alpha-\lambda^k s^k\big\|_2\ .\end{aligned}$$

In the last step we have used that the solution of the minimization of the linear function over the Euclidean unit ball is given by

$$u^* = -\frac{A\alpha-\lambda^k s^k}{\|A\alpha-\lambda^k s^k\|_2},$$

if $\big\|A\alpha-\lambda^k s^k\big\| \neq 0$ and otherwise u^* is an arbitrary element of the Euclidean unit ball. Transforming the maximization problem into a minimization problem finishes the proof of the first statement. Regarding the gradient, a straightforward computation shows that

$$\begin{aligned}(\nabla\Psi(\alpha))_{rs} &= \sum_{i\in V} 2\Big((A\alpha)_i-\lambda^k s_i^k\Big)\cdot\Big(\tfrac{1}{2}w_{is}\delta_{i=r}-\tfrac{1}{2}w_{ir}\delta_{i=s}\Big)\\ &= w_{rs}\left(\Big((A\alpha)_r-\lambda^k s_r^k\Big)-\Big((A\alpha)_s-\lambda^k s_s^k\Big)\right).\end{aligned}$$

Thus, regarding the Lipschitz constant, we obtain for $\alpha,\alpha'\in\mathbb{R}^{|E|}$,

$$\begin{aligned}\big\|\nabla\Psi(\alpha)-\nabla\Psi(\alpha')\big\|_2^2 &= \sum_{(r,s)\in E} w_{rs}^2\Big((A\alpha)_r-(A\alpha')_r-(A\alpha)_s+(A\alpha')_s\Big)^2\\ &\leq 2\sum_{(r,s)\in E} w_{rs}^2\left((A\alpha)_r-(A\alpha')_r\right)^2+2\sum_{(r,s)\in E} w_{rs}^2\left((A\alpha)_s-(A\alpha')_s\right)^2\\ &= 4\sum_{(r,s)\in E} w_{rs}^2\left((A\alpha)_r-(A\alpha')_r\right)^2,\end{aligned}$$

where we used the fact that for $a,b\in\mathbb{R}$ it holds that $(a-b)^2\leq 2a^2+2b^2$, as well as the symmetry of W. This can be further rewritten as

$$\begin{aligned}&\sum_{(r,s)\in E} w_{rs}^2\Big(\sum_{j\,|\,(r,j)\in E} w_{rj}(\alpha_{rj}-\alpha'_{rj})-(\alpha_{jr}-\alpha'_{jr})\Big)^2\\ \leq\ &2\sum_{(r,s)\in E} w_{rs}^2\Big(\sum_{j\,|\,(r,j)\in E} w_{rj}(\alpha_{rj}-\alpha'_{rj})\Big)^2+2\sum_{(r,s)\in E} w_{rs}^2\Big(\sum_{j\,|\,(r,j)\in E} w_{rj}(\alpha_{jr}-\alpha'_{jr})\Big)^2\end{aligned}$$

$$\leq 2 \sum_{(r,s)\in E} w_{rs}^2 \sum_{j\,|\,(r,j)\in E} w_{rj}^2 \sum_{i\,|\,(r,i)\in E} (\alpha_{ri} - \alpha'_{ri})^2 + 2 \sum_{(r,s)\in E} w_{rs}^2 \sum_{j\,|\,(r,j)\in E} w_{rj}^2 \sum_{i\,|\,(r,i)\in E} (\alpha_{ir} - \alpha'_{ir})^2$$

$$\leq 2 \left(\max_r \sum_{s\,|\,(r,s)\in E} w_{rs}^2\right)^2 \sum_{(r,i)\in E} (\alpha_{ri} - \alpha'_{ri})^2 + 2 \left(\max_r \sum_{s\,|\,(r,s)\in E} w_{rs}^2\right)^2 \sum_{(r,i)\in E} (\alpha_{ir} - \alpha'_{jr})^2,$$

where we used the Cauchy-Schwarz inequality in the second step. Finally,

$$\left\|\nabla\Psi(\alpha) - \nabla\Psi(\alpha')\right\|_2^2 \;\leq\; 4 \left(\max_r \sum_{s\,|\,(r,s)\in E} w_{rs}^2\right)^2 \left\|\alpha - \alpha'\right\|_2^2,$$

thus the Lipschitz constant is upper bounded by $2 \max_r \sum_{s\,|\,(r,s)\in E} w_{rs}^2$. □

In contrast to the primal problem, the objective of the dual problem is smooth. It can be efficiently solved using Nesterov's fast projected gradient method [Nesterov, 1983, Beck and Teboulle, 2009], see Section 6. The only input is an upper bound on the Lipschitz constant of the gradient of the objective, which is given in Lemma 7.38.

Note that in the algorithm can be implemented efficiently with a memory requirement of α equal to the number of nonzero entries of W. Moreover, it can be further optimized as follows: Note that we can write $\forall i, j \in V$, $\alpha_{ij} - \alpha_{ji} = \frac{1}{2}(\alpha_{ij} - \alpha_{ji}) - \frac{1}{2}(\alpha_{ji} - \alpha_{ij})$. Thus, at each step of the algorithm, we can replace the variable α by the variable $\widehat{\alpha}$ defined as $\widehat{\alpha}_{ij} := \frac{1}{2}(\alpha_{ij} - \alpha_{ji})$, as $\|\widehat{\alpha}\|_\infty \leq 1$ and α and $\widehat{\alpha}$ achieve the same objective value of the inner problem. However, in contrast to α, the vector $\widehat{\alpha}$ is anti-symmetric, i.e. $\widehat{\alpha}_{ij} = -\widehat{\alpha}_{ji}$. Thus in an efficient implementation we only need to consider the upper triangular part of $\widehat{\alpha}$, the remaining entries can be hard-coded in the implementation, thus reducing the memory requirement by a factor 2.

Thus the most expensive part of each iteration of the algorithm is a sparse matrix multiplication, which scales linearly in the number of edges. Nesterov's method provides a good solution in a few steps which guarantees descent in functionals $Q_1^{(\mathrm{u})}$ or $Q_1^{(\mathrm{n})}$ and thus makes the nonlinear IPM very fast. The resulting algorithm is shown in Alg. 16. Here, $P_{B_\infty(1)}$ denotes the projection on the L_∞ unit ball, given as $B_\infty(1) := \{x \in \mathbb{R} \,|\, |x| \leq 1\}$.

7.5 Symmetric vertex expansion

In this section we consider a variant of the above balanced partition problems based on the *vertex expansion* of a set S. The minimum vertex expansion

16 Solution of the dual inner problem with Nesterov's method

Input: Lipschitz-constant L of $\nabla\Psi$,
Initialization: $\theta^0 = 1$, $\alpha^0 \in \mathbb{R}^{|E|}$,
repeat
  $z^t = A\alpha^t - \lambda^k s^k$
  $\beta_{rs}^{t+1} = P_{B_\infty(1)}\left(\alpha_{rs}^t - \frac{1}{L} w_{rs}\left(z_r^t - z_s^t\right)\right)$
  $\theta^{t+1} = \frac{1+\sqrt{1+4(\theta^t)^2}}{2}$,
  $\alpha_{rs}^{t+1} = \beta_{rs}^{t+1} + \frac{\theta^t - 1}{\theta^{t+1}}\left(\beta_{rs}^{t+1} - \beta_{rs}^t\right)$.
until duality gap $< \epsilon$

of a set C is given as (see e.g. Hoory et al. [2006])

$$\min_{|C| \leq \frac{V}{2}} \frac{|\mathrm{N}(C)|}{|C|},$$

where $\mathrm{N}(C)$ denotes the set of vertices in $V \backslash C$ which are adjacent to C. Note that the above criterion is not symmetric, as in general we do not have $\mathrm{N}(C) = \mathrm{N}(\overline{C})$. Since we are interested in obtaining a partition of the graph, we therefore consider a variant of the above problem called the *symmetric vertex expansion*, given as (see e.g. Louis et al. [2013])

$$\min_{C \subset V} \frac{\left|\mathrm{N}(C) \cup \mathrm{N}(\overline{C})\right|}{\min\{|C|, |\overline{C}|\}} =: \mathrm{VE}(C, \overline{C}). \tag{7.13}$$

A normalized variant of the above problem is given as

$$\min_{C \subset V} \frac{\left|\mathrm{N}(C) \cup \mathrm{N}(\overline{C})\right|}{\min\{\mathrm{vol}(C), \mathrm{vol}(\overline{C})\}} =: \mathrm{NVE}(C). \tag{7.14}$$

Thus, the difference to the ratio Cheeger cut RCC and normalized Cheeger cut NCC considered in the previous sections (see Eq. 7.2 and Eq. 7.4) is that while in the case of the Cheeger cut we consider the sum of the edge weights between the two clusters, in the case of the symmetric vertex expansion we consider the number of vertices involved in the cut.

While the criterion looks similar to the Cheeger cut criterion from the last section, it often behaves very differently. In Figure 7.7 we give an example where optimizing vertex expansion and Cheeger cut lead to completely different clusters. In this example, we have four fully connected subgraphs of k nodes each ($k > 8$), some of which have connections to the other connected components (here we draw only the nodes which have connections to the other parts of the graph).

We first consider the problem of finding the optimal ratio Cheeger cut of the graph. Cutting the graph horizontally, i.e. separating the fully connected

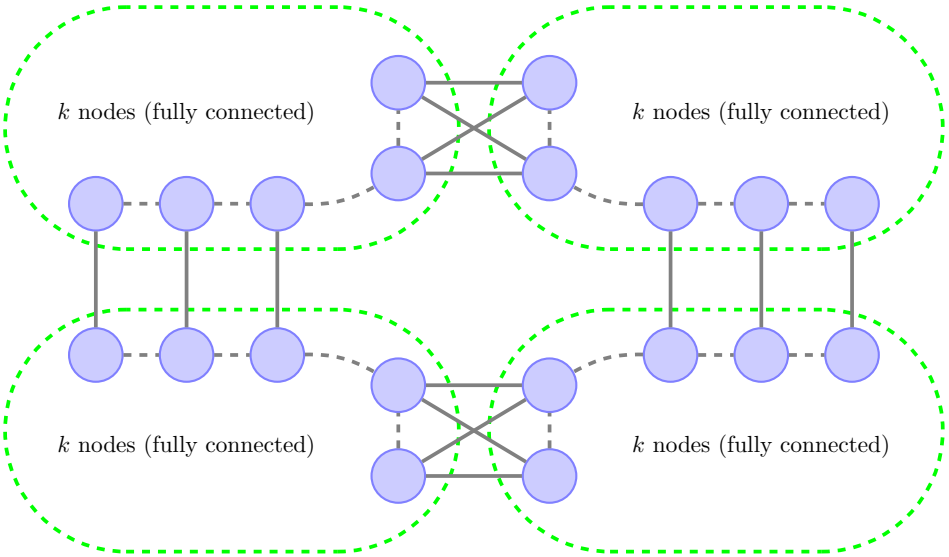

Figure 7.7: Example where optimizing symmetric vertex expansion and Cheeger cut produce completely different results. The optimal Cheeger cut cuts horizontally, while the optimal vertex expansion cuts vertically.

graphs on the top from the ones on the bottom, cuts 6 vertices and leads to a balanced partition of $2k$ vertices in each cluster. Thus the Cheeger cut is $\frac{3}{k}$. On the other hand, cutting the graph vertically leads to a Cheeger cut of $\frac{8}{2k} = \frac{4}{k}$. Moreover, separating one of the connected components leads to a Cheeger cut of $\frac{7}{k}$. Finally, cutting inside any of the fully connected components leads to a cut of at least k, while the denominator (balancing term) is at most $2k$, and thus the Cheeger cut is lower bounded by $\frac{1}{2}$, which is strictly larger than $\frac{3}{k}$, since $k > 8$. Thus the optimal Cheeger cut is given by the horizontal cut.

On the other hand, when considering the number of involved vertices in the cut, instead of the sum of the edge weights, the results are different. In the case of the horizontal cut, 12 vertices are involved in the cut, leading to a vertex expansion of $\frac{6}{k}$, whereas only 8 are involved in the vertical cut, leading to a vertex expansion of $\frac{4}{k}$. Moreover, separating one of the fully connected components leads to a vertex expansion of $\frac{10}{k}$. Finally, cutting inside one of the fully connected components involves at least k vertices and thus with the same argument as before leads to a vertex expansion of at least $\frac{1}{2}$. Thus, since $k > 8$, the optimal vertex expansion is achieved by the vertical cut.

An important difference between Cheeger cut and symmetric vertex expansion is that the latter is invariant to changes in the weights between vertices, as well as addition of edges between nodes which are already at

the boundary between the two sets. Thus, optimizing symmetric vertex expansion is suitable for partitioning problems for instance in communication networks where the cost is associated with the number of nodes involved in the cut, rather than the total edge weight of the cut itself.

7.5.1 Tight relaxation of symmetric vertex expansion

In this section, we give the tight relaxation of the symmetric vertex expansion problems in (7.13) and (7.14). In contrast to before, we do not specify the Lovász extension of the numerator explicitly. The reason is that an explicit form of the Lovász extension will not be necessary to solve the inner problem appearing in RatioDCA, as described in Chapter 6.

Instead, we will make use of Lemma 2.20 and express the Lovász extension implicitly in terms of its subgradient, which can be computed directly using only function evaluations of the original set objective. In the next section we will then solve the inner problem efficiently using the bundle-level method discussed in Section 6.5, which requires only the evaluation of the subgradient in each step.

We now state the tight relaxation for the symmetric vertex expansion problems. We use again the median and weighted median, see Eq. (7.7), as well as the sets $C_i := \{j \in V | f_j \geq f_i\}$ for all $i = 1, \ldots, n$.

Theorem 7.39 (Tight relaxation of symmetric vertex expansion). *It holds that*

$$\min_{C \subset V} \frac{\left|\mathrm{N}(C) \cup \mathrm{N}(\overline{C})\right|}{\min\{|C|, |\overline{C}|\}} = \min_{f \in \mathbb{R}^n} \frac{\langle r(f), f \rangle}{\|f - \mathrm{median}(f)\mathbf{1}\|_1} \quad \textit{and}$$

$$\min_{C \subset V} \frac{\left|\mathrm{N}(C) \cup \mathrm{N}(\overline{C})\right|}{\min\{\mathrm{vol}(C), \mathrm{vol}(\overline{C})\}} = \min_{f \in \mathbb{R}^n} \frac{\langle r(f), f \rangle}{\|f - \mathrm{median}_d(f)\mathbf{1}\|_{1,d}},$$

where $r(f)_i = |\mathrm{N}(C_i)| - |\mathrm{N}(C_{i+1})| + \left|\mathrm{N}(\overline{C}_i)\right| - \left|\mathrm{N}(\overline{C}_{i+1})\right|$, *for all* $i = 1, \ldots, n$.

Proof. By Lemma 2.20, the Lovász extension of $\left|\mathrm{N}(C) \cup \mathrm{N}(\overline{C})\right|$ is given as $\langle r(f), f \rangle$, where $r(f)_i = \left|\mathrm{N}(C_i) \cup \mathrm{N}(\overline{C_i})\right| - \left|\mathrm{N}(C_{i+1}) \cup \mathrm{N}(\overline{C_{i+1}})\right|$. Using the fact that $N(C)$ and $N(\overline{C})$ are disjunct, we obtain the above expression for $r(f)$. The Lovász extension of the denominator has been shown in Lemma 7.22. We can now apply Theorem 4.2. Finally, using the fact that the ratio is symmetric, we can replace optimization over $\mathbb{R}^n_+$ by optimization of $\mathbb{R}^n$, which yields the result. □

7.5.2 Solution via nonlinear inverse power method

As before, we use the nonlinear IPM from Section 5.3 to optimize the ratio derived in Theorem 7.39. However, in contrast to before, we replace the

2-norm constraint $\|f\|_2 \leq 1$ in the inner problem by an ∞-norm constraint $\|f\|_\infty \leq 1$ (see the discussion in Section 5.3). The inner problem solved in each step k of the nonlinear IPM now has the form

$$\min_{\|f\|_\infty \leq 1} \langle r(f) - \lambda^k s^k, f \rangle,$$

where $r(f)$ is an element of the subdifferential of the numerator (according to Theorem 7.39), and s^k is an element of the subdifferential of the denominator at step k (which is constant for the inner problem at step k). Since the inner problem has the general form in 6.16, we can solve it using the bundle-level method discussed in Section 6.5.

Alg. (17) shows the resulting algorithm for the (unnormalized) symmetric vertex expansion problem. Analogously one obtains the algorithm in the normalized case. Similarly to before, the general convergence properties of the nonlinear IPM imply the following results for Alg. 17, which are direct corollaries of the results in Prop. 5.9, Theorem 5.10 and Theorem 5.13. We omit the exact form of the nonlinear eigenproblem associated to the problems in (7.13) and (7.14).

17 Optimizing the symmetric vertex expansion

Input: weight matrix W
Initialization: nonconstant f^0 with $\left\|f^0\right\|_\infty \leq 1$, accuracy ϵ
repeat
  $g^k = f^k - \text{median}\left(f^k\right)\mathbf{1}$
  $s_i^k = \begin{cases} \text{sign}(g_i^k), & \text{if } g_i^k \neq 0, \\ -\frac{|g_+^k| - |g_-^k|}{|g_0^k|}, & \text{if } g_i^k = 0 \end{cases}$
  $f^{k+1} = \arg\min_{\|f\|_\infty \leq 1} \langle r(f) - \lambda^k s^k, f \rangle$
  $\lambda^{k+1} = Q_1^{(\mathrm{u})}(f^{k+1})$
until $\frac{\left|\lambda^{k+1} - \lambda^k\right|}{\lambda^k} < \epsilon$

Theorem 7.40 (Convergence). *Let Q denote the functional in Theorem 7.39. The sequence f^k produced by Alg. 17 satisfies $Q(f^k) > Q(f^{k+1})$ for all $k \geq 0$, or the sequence terminates. Moreover, f^k has a subsequence converging to a solution of the nonlinear eigenproblem associated to Q.*

Theorem 7.41 (Vertex expansion improvement). *Let C be any feasible set, f denote the result of Alg. 17 after initializing with $\frac{1}{|C|}\mathbf{1}_C$, and C_f be the set obtained by optimal thresholding of f. Either Alg. 17 terminates after one iteration, or the set C_f is feasible and it holds that* $\mathrm{VE}(C, \overline{C}) > \mathrm{VE}(C_f, \overline{C_f})$.

7.6 Multi-partitioning

Up to now we have only considered bi-partitions into two sets C and $\overline{C}$. Usually one is interested in computing a multi-partition of the graph into sets $(C_1, \ldots, C_k)$, where $\cup_{i=1,\ldots,k} C_i = V$. To avoid overlapping clusters, typically one also requires $\cap_{i=1,\ldots,k} C_i = \emptyset$.

A generalization of the ratio cut criterion to the multi-partition case is given as [von Luxburg, 2007],

$$\mathrm{RCut}(C_1, \ldots, C_k) = \sum_{i=1}^{k} \frac{\mathrm{cut}(C_i, \overline{C_i})}{|C_i|},$$

and analogously for the normalized cut. Note that there seems to be no generally accepted multi-partition version of the Cheeger cut objective. Interpreting the RCut as L_1 norm and the RCC as L_∞ norm, a straight-forward way to define a multi-partition version of the ratio Cheeger cut is given as

$$\mathrm{RCC}(C_1, \ldots, C_k) = \max_{i \in \{1 \ldots k\}} \frac{\mathrm{cut}(C_i, \overline{C_i})}{|C_i|},$$

and similarly for the normalized Cheeger cut. Another generalization of the Cheeger cut is given by the objective (see Luo et al. [2010])

$$\mathrm{RCC}'(C_1, \ldots, C_k) = \tfrac{1}{k} \sum_{i=1}^{k} \frac{\mathrm{cut}(C_i, \overline{C_i})}{\min_{i \in \{1 \ldots k\}} |C_i|},$$

Yet another different generalization of the Cheeger cut is given as (see Bresson et al. [2013])

$$\mathrm{RCC}''(C_1, \ldots, C_k) = \tfrac{1}{k} \sum_{i=1}^{k} \frac{\mathrm{cut}(C_i, \overline{C_i})}{\min\left\{|C_i|, |\overline{C_i}|\right\}}.$$

Similarly, one could define generalizations of the symmetric vertex expansion in (7.13) and (7.14). Since it is unclear which of the above generalizations is the best choice for a given application, we focus in this thesis on the established multi-cut variants of the RCut and NCut criterion. To compute a multi-partition into k clusters, we use a greedy recursive bi-partitioning scheme, i.e. in each step, we partition each cluster tentatively and then choose to keep the partition yielding a better objective in the multi-cut criterion. Clusters are split consecutively until the desired number of clusters is reached.

A different approach used in standard spectral clustering is to compute the set of k eigenvectors of the unnormalized/normalized graph Laplacian, which can be seen as a relaxation of the multi-cut version of the NCut criterion [von Luxburg, 2007]. In the new k-dimensional representation defined

by the first k eigenvectors, one then applies a standard clustering algorithm, for instance k-means [MacQueen, 1967]. This approach usually yields better cuts in terms of the multi-cut objective than the recursive splitting scheme.

After the initial publication of p-spectral clustering in [Bühler and Hein, 2009a], a generalization to multiple eigenvectors was proposed by Luo et al. [2010]. Their main observation was that given two eigenvectors f and g of the graph Laplacian $\Delta_p^{(\mathrm{u})}$ with eigenvectors λ_f and λ_g, one observes that they satisfy *approximately* (up to the second order Taylor expansion),

$$\langle \phi_p(f), \phi_p(g) \rangle \approx 0,$$

a property referred to as *p-orthogonality* which generalizes the fact that $\langle \phi_p(f), \mathbf{1} \rangle = 0$ as shown in Bühler and Hein [2009b]. Luo et al. [2010] then showed that under the assumption that the above property is fulfilled *exactly*, a set of k eigenvectors with different eigenvalues is given as the local optimal solution of the problem

$$\begin{aligned} \min_F \;\; & J(F) = \sum_{i=1}^{k} \overline{Q}_p\big(f^{(i)}\big) \\ \text{subject to} \;\; & \big\langle \phi_p\big(f^{(i)}\big), \phi_p\big(f^{(j)}\big) \big\rangle = 0, \;\; \forall i \neq j, \end{aligned}$$

where $F = [f^{(1)}, \ldots, f^{(k)})] \in \mathbb{R}^{n \times k}$. The set of feasible variables F can be seen as a generalization of the Stiefel manifold [Absil et al., 2008] to p-orthogonal functions. Since the above problem is intractable, the feasible set was then replaced by the Stiefel manifold (i.e. the set of variables $F \in \mathbb{R}^{n \times k}$ such that $F^T F = I$) and the problem was solved using standard manifold optimization techniques [Absil et al., 2008]. Then it was argued that the result gives a good approximate solution of the original problem. Given the higher-dimensional embedding defined by the solution of their method, they then use a standard clustering algorithm such as K-means, as in the usual approach in standard spectral clustering.

While in their experiments their method was shown to perform well in practice (in particular outperformed p-spectral clustering with $p = 1.2$ on several datasets), it is unclear what is the exact connection to the eigenvectors of the graph p-Laplacian. Moreover, it is also unclear how the relaxation of Luo et al. [2010] relates to the optimum of the original multi-cut objective.

Recently, Bresson et al. [2013] proposed an algorithm for total variation based multi-class clustering as follows. They considered the problem of minimizing the following variant of the Cheeger multi-cut problem,

$$\sum_{i=1}^{k} \frac{\operatorname{cut}(C_i, \overline{C_i})}{\min\big\{\lambda |C_i|, |\overline{C_i}|\big\}},$$

over all disjoint partitions $(C_1, \ldots, C_k)$ of the vertex set. The parameter λ is used to control the balance between set sizes. Setting $\lambda = k - 1$ ensures

that the maximum of each denominator (viewed separately) is achieved if $|C_i| = \frac{n}{k}$, thus in total it leads to a bias towards balanced partitions. The problem was first transformed into the equivalent problem

$$\begin{aligned} \min_{f^{(i)} \in \{0,1\}^n,\, i=1\ldots k} \quad & \sum_{i=1}^{k} \frac{\frac{1}{2}\sum_{r,s\in V} w_{rs}|f_r^{(i)} - f_s^{(i)}|}{\left\|f^{(i)} - \text{median}_\lambda(f^{(i)})\mathbf{1}\right\|_{1,\lambda}} \\ \text{subject to} \quad & \sum_{i=1}^{k} f^{(i)} = \mathbf{1}, \end{aligned} \tag{7.15}$$

where the λ-median is the $k+1$-st largest value in f, for $k = \lfloor n/(\lambda+1) \rfloor$, and here $\|f\|_{1,\lambda}$ denotes the asymmetric L_1 norm,

$$\|f\|_{1,\lambda} := \sum_{i=1}^{n} |f_i|_\lambda\,, \qquad \text{where } |t|_\lambda = \begin{cases} \lambda t, & \text{if } t \geq 0 \\ -t & \text{if } t < 0 \end{cases}.$$

The problem was then relaxed by replacing the constraint $f^{(i)} \in \{0,1\}^n$ by $f^{(i)} \in [0,1]^n$. However, note that this is not a tight relaxation. Moreover, note that while in (7.15) the simplex constraint together with the fact that the vectors $f^{(i)}$ have only values 0 and 1 implies that the vectors $f^{(i)}$ are mutually orthogonal, this is not the case for the relaxed version anymore. The authors then developed a proximal splitting scheme to optimize the resulting continuous objective. In the experiments we compare the obtained results against the results by our method.

Since at the moment we are not able to compute higher eigenvectors of the graph 1-Laplacian, we perform the recursive multi-partitioning discussed above. However, we will show in the experiments that 1-spectral clustering with the above recursive splitting scheme still outperforms spectral clustering when using the higher eigenvectors.

7.7 Experimental results

In all experiments, given some data $x_1, \ldots, x_n$, we first construct a symmetric K-NN graph $G(V, E, W)$ representing the similarity between data points, i.e. each vertex in V corresponds to a data point and two vertices i and j are connected if point x_i is among the K nearest neighbors of point x_j or vice versa. We choose $K = 10$ and determine the neighborhood of a point according to the Euclidean distance.

The weights are usually chosen to reflect the similarity between the corresponding points. A common approach is to use Gaussian weights

$$w_{ij} = e^{-\frac{1}{\sigma^2}\|x_i - x_j\|^2},$$

where the parameter σ controls the width of the Gaussian, and thus should be chosen in such a way that it reflects the scale of the given data. In order to

be able to cope with data with different local scales, we adopt an approach similar to the one in Zelnik-Manor and Perona [2004], where instead of choosing a single scaling parameter σ, we compute for each point x_i a local scaling parameter σ_i which we set to a factor α times the Euclidean distance of x_i to its K-nearest neighbor. In the experiments, we chose $\alpha = \frac{1}{2}$. The weights w_{ij} are then computed as

$$w_{ij} = \max\{\mathrm{s}_i(j), \mathrm{s}_j(i)\}, \text{ where } \mathrm{s}_i(j) = e^{-\frac{1}{\sigma_i^2}\|x_i - x_j\|^2}.$$

Since we are applying the methods to datasets with known class structure, we can use this information to evaluate the quality of the found clusterings by checking the agreement with the true class structure. Thus, for a dataset with known number of classes k the data is first clustered into k clusters. Then, in order to evaluate the clustering, we treat our clustering problem as a classification problem and use the obtained clustering to predict a 'label' for each point. The label of each cluster is determined via a majority vote. We now define our error measure as the relative number of times the 'predicted' label disagrees with the 'true' label. This leads us to the following error measure

$$\mathrm{error}(C_1, .., C_k) = \frac{1}{|V|} \sum_{i=1}^{k} \sum_{j \in C_i} \delta_{Y_j \neq Y_i'}, \tag{7.16}$$

where for a given vertex $j \in C_i$, Y_j denotes the true label of j and Y_i' is the dominant label in cluster C_i. Thus the above error measure quantifies the agreement of the found clusters $C_1, \ldots, C_k$ with the class structure.

7.7.1 High-dimensional noisy two moons

As in Bühler and Hein [2009a], the two moons dataset is generated as two half-circles in $\mathbb{R}^2$ embedded into a d-dimensional space where Gaussian noise $\mathcal{N}(0, \sigma^2 \mathbf{1}_d)$ is added. In Fig. 7.8, we show the edge structure of the resulting graph for $d = 100$, $n = 2000$ and $\sigma^2 = 0.02$. Note that this dataset is far from being trivial since the high-dimensional noise has corrupted the graph.

We perform balanced graph partitioning according to the normalized Cheeger cut criterion on the high-dimensional two moons dataset with the goal of separating the two half-circles. We compare the performance of normalized spectral clustering ($p = 2$), normalized p-spectral clustering for different values of p with $1 < p < 2$ as presented in Bühler and Hein [2009a], as well as 1-spectral clustering, i.e. the inverse power method applied to the tight relaxation of the normalized Cheeger cut criterion ($p = 1$) proposed in this chapter. In the case of the inverse power method, we use the best result of 10 runs with random initializations and one run initialized with the second eigenvector of the normalized graph Laplacian.

In Fig. 7.8 we plot the values of NCC, the eigenvalue of the graph p-Laplacian, NCut as well as the error. The explicit values are given in Table

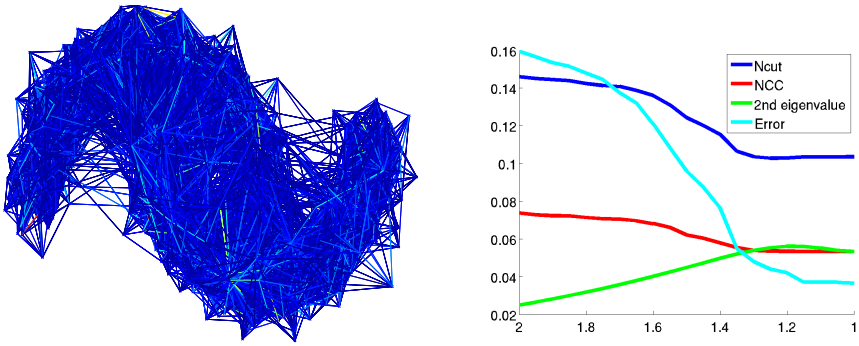

Figure 7.8: *Left:* Plot of the edge structure of the two moons data set, 2000 points in 100 dimensions, noise variance 0.02. *Right:* Values of NCC, the eigenvalue $\lambda_p^{(2)}$, NCut and the error obtained for different values of p.

7.1. One observes that for decreasing values of p, the NCC values and error values decrease. The best error is achieved for $p = 1$. Moreover, we observe that (as predicted by Theorem 7.34) the second eigenvalue of the 1-Laplacian is equal to the obtained Cheeger cut. Note that also for $p > 1$ we always report the NCC and NCut criteria and not the corresponding RCC_p criterion. Thus the curve for the NCC criterion is slightly below the corresponding functional value for $p = 1.2$ and $p = 1.1$. One easily checks that the value for the second eigenvalue is still within the bounds given by the generalized isoperimetric inequality (Theorem 7.20).

Note that we have optimized the NCC criterion for each method up to now. Thus, while we see a significant decrease also in the NCut criterion, the best NCC partition does not correspond to the best NCut partition in this case. We repeat the experiment with the method based on the tight relaxation of the NCut criterion. As expected, in this case we obtain an even better NCut value. Moreover, analogously to the result for the NCC criterion, the objective value at the optimal point (which corresponds to the eigenvalue of the corresponding nonlinear eigenproblem) is equal to the obtained NCut value. Note that we obtain a slightly higher error value, thus optimizing NCC seems to be the better choice for this dataset.

In the left column of Fig. 7.9 we show the eigenvector of the graph p-Laplacian for $p = 2.0, 1.4, 1.2, 1.1, 1.0$. In the middle column, the values of the eigenvector have been sorted in increasing order. For $p = 2$, the values are spread over the interval, whereas for decreasing values of p, they concentrate on two peaks. The third column in Fig. 7.9 shows the corresponding clusters found by optimal thresholding according to the NCC criterion. For $p \to 1$, the clustering is almost perfect despite the difficulty of this dataset.

We repeat the experiment for the ratio Cheeger cut (RCC), comparing the performance of standard unnormalized spectral clustering ($p = 2$), un-

Table 7.1: Results of graph partitioning on the high dimensional noisy two moons dataset according to NCC and NCut criteria.

Method	NCC	NCut	Error	Eigenvalue
Standard spectral	0.0737	0.1461	0.1595	0.0250
p-spectral ($p = 1.9$)	0.0723	0.1445	0.1535	0.0282
p-spectral ($p = 1.8$)	0.0714	0.1424	0.1480	0.0318
p-spectral ($p = 1.7$)	0.0705	0.1408	0.1375	0.0358
p-spectral ($p = 1.6$)	0.0681	0.1361	0.1210	0.0401
p-spectral ($p = 1.5$)	0.0621	0.1242	0.0960	0.0449
p-spectral ($p = 1.4$)	0.0579	0.1155	0.0765	0.0497
p-spectral ($p = 1.3$)	0.0540	0.1036	0.0480	0.0539
p-spectral ($p = 1.2$)	0.0534	0.1029	0.0420	0.0561
p-spectral ($p = 1.1$)	0.0533	0.1035	0.0370	0.0551
1-spectral (Tight NCC)	**0.0533**	0.1036	**0.0365**	0.0533
1-spectral (Tight NCut)	0.0538	**0.1024**	0.0405	0.1024

normalized p-spectral clustering ($p = 1.1$) as well as 1-spectral clustering (tight relaxation of RCC). Moreover, we evaluate the TV-based method for RCC minimization of Szlam and Bresson [2010]. In the case of the IPM, we use the best result of 10 runs with random initializations and one run initialized with the second eigenvector of the unnormalized graph Laplacian. For the method of Szlam and Bresson [2010] we use the normalized graph Laplacian as proposed by the authors and add 10 random runs.

Table 7.2 shows the average RCC and error for 100 draws of a two-moons dataset with 2000 points using the same parameters as above. IPM and the TV-based method yield similar results, slightly better than 1.1-spectral and clearly outperforming standard spectral clustering. In terms of runtime, inverse power method and Szlam and Bresson [2010] are in the same order of magnitude ($\sim$ 5 seconds per run).

Table 7.2: Results of graph partitioning on the high dimensional noisy two moons dataset according to RCC criterion.

Method	Avg. RCC	Avg. error
Standard spectral	0.0247 ($\pm$ 0.0016)	0.1685 ($\pm$ 0.0200)
p-spectral ($p = 1.1$)	0.0196 ($\pm$ 0.0016)	0.0578 ($\pm$ 0.0285)
Szlam and Bresson [2010]	**0.0195 ($\pm$ 0.0015)**	0.0491 ($\pm$ 0.0181)
1-spectral (Tight RCC)	**0.0195 ($\pm$ 0.0015)**	**0.0462 ($\pm$ 0.0161)**

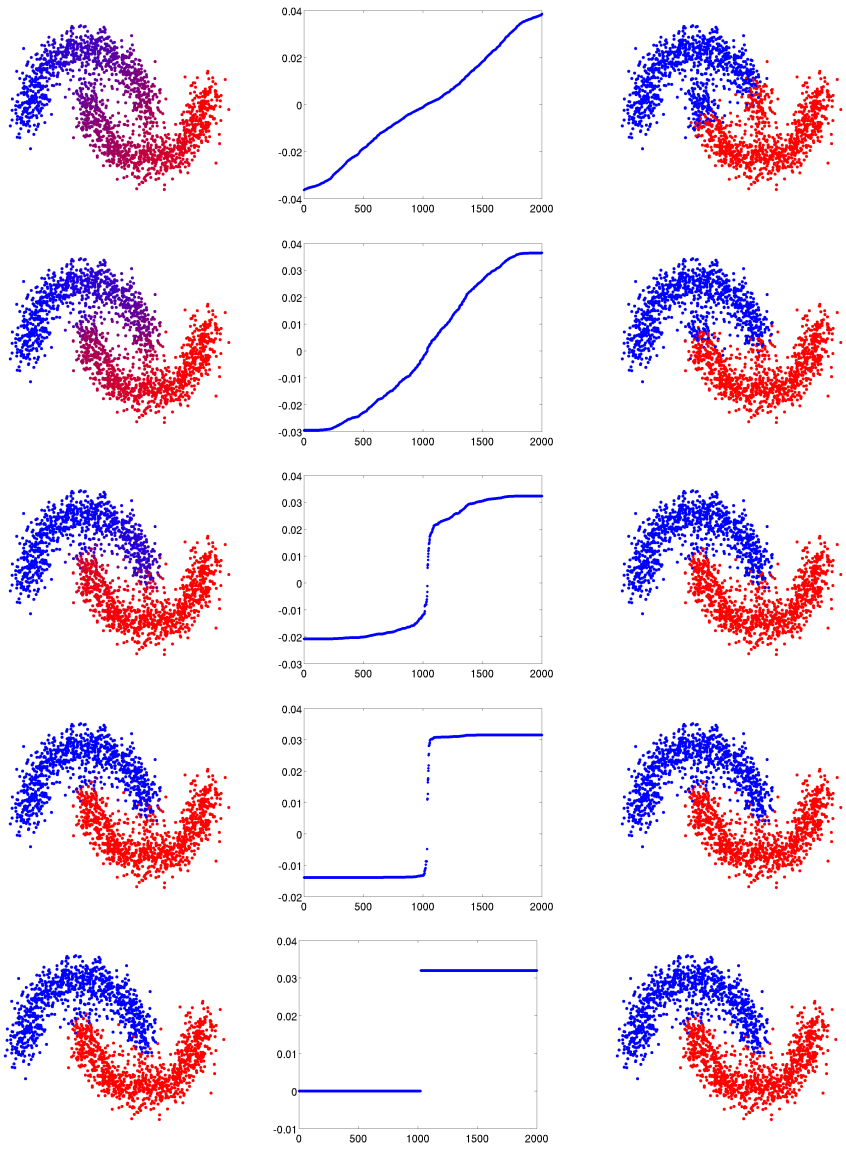

Figure 7.9: Results for the two moons data set, 2000 points in 100 dimensions, noise variance 0.02. *First column, from top to bottom:* Second eigenvector of the p-Laplacian for $p = 2.0, 1.4, 1.2, 1.1, 1.0$. *Second column:* Values of the second eigenvector sorted in increasing order. *Last row:* Resulting clustering after finding optimal threshold according to the NCC criterion.

7.7.2 Graph partitioning benchmark

In this experiment we evaluate our method on several graphs downloaded from the graph partitioning archive [Soper et al., 2004, Walshaw]. This archive consists of several unweighted sparse graphs of medium to large size. We evaluate our method on a subset of the available datasets with different sizes and levels of sparsity, including the two largest datasets available in the benchmark. In Table 7.3 we summarize the properties of the used datasets.

Table 7.3: Benchmark datasets from the graph partitioning archive [Walshaw] used in the experiment.

Dataset	$\|V\|$	$\|E\|$	Avg. Degree
add20	2395	7462	6.2
data	2851	15093	10.6
3elt	4720	13722	5.8
whitaker3	9800	28989	5.9
cs4	22499	43858	3.9
t60k	60005	89440	3.0
finan512	74752	261120	7.0
fe_ocean	143437	409593	5.7
m14b	214765	1679018	15.6
auto	448695	3314611	14.8

Our goal is to find the bi-partition of the graphs achieving the optimal RCC value. We compare the result obtained by 1-spectral clustering (tight relaxation of RCC criterion) to standard unnormalized spectral clustering, as well as unnormalized p-spectral clustering for $p = 1.1$. Next we repeat the experiment for the NCC criterion, using the normalized variants of the above methods.

The results are summarized in Table 7.4 for both experiments, where we report the obtained Cheeger cut values (RCC or NCC) as well as the total runtime in seconds. In order to demonstrate our quality guarantee (Theorem 7.37), in the case of 1-spectral clustering we first perform one run initialized with the thresholded second eigenvector of the graph Laplacian. The results are shown in the third column. In the fourth column, we add 100 random initializations.

Table 7.4: Obtained RCC and NCC values as well as total runtime in seconds on several datasets from the graph partitioning archive. In the case of 1-Spectral clustering, the third column denotes the result of one run initialized with the solution of the standard spectral relaxation. In the fourth column, we add 100 random initializations. 1-Spectral clustering consistently outperforms the other methods in terms of obtained Cheeger cut values.

	Standard Spectral		p-Spectral ($p = 1.1$)		1-Spectral (init spectral)		1-Spectral (+100 runs)	
	RCC	Time	RCC	Time	RCC	Time	RCC	Time
add20	0.1667	2	**0.1579**	142	**0.1579**	3	**0.1579**	53
data	**0.02857**	2	**0.02857**	112	**0.02857**	4	**0.02857**	76
3elt	0.04246	2	0.03856	407	0.0376	6	**0.03747**	84
whitaker3	0.02703	5	0.02654	550	0.02653	12	**0.02576**	166
cs4	0.04041	3	0.0336	1907	0.03331	13	**0.03286**	334
t60k	0.002723	451	0.002232	11604	**0.002069**	769	**0.002069**	1583
finan512	**0.004334**	339	**0.004334**	4735	**0.004334**	571	**0.004334**	2618
fe_ocean	0.007701	253	0.004516	19262	**0.004436**	423	**0.004436**	3722
m14b	0.03727	287	0.03621	50682	0.03576	948	**0.03572**	29368
auto	0.04834	761	0.04657	96692	0.04618	2196	**0.04431**	42397

	Standard Spectral		p-Spectral ($p = 1.1$)		1-Spectral (init spectral)		1-Spectral (+100 runs)	
	NCC	Time	NCC	Time	NCC	Time	NCC	Time
add20	0.07407	0	**0.06667**	218	**0.06667**	1	**0.06667**	45
data	**0.002714**	2	**0.002714**	343	**0.002714**	4	**0.002714**	75
3elt	0.007267	3	0.006692	556	0.006471	13	**0.006418**	85
whitaker3	0.004568	7	0.004487	513	0.004485	15	**0.004355**	169
cs4	0.01045	5	0.008688	3087	0.008614	16	**0.008474**	337
t60k	0.0009201	705	0.0007624	12989	**0.0006939**	1224	**0.0006939**	1933
finan512	0.0006613	116	0.0006613	5377	**0.0006204**	236	**0.0006204**	2400
fe_ocean	0.001346	348	0.0008312	25452	**0.0007822**	563	**0.0007822**	3936
m14b	0.002385	214	0.002319	23750	0.002288	787	**0.002285**	29348
auto	0.003256	576	0.003139	93306	0.003105	1990	**0.002986**	46834

For both RCC and NCC criterion, we observe that standard spectral clustering is significantly outperformed by the other methods in terms of obtained Cheeger cut value. Moreover, we see that in most cases, performing one run of 1-spectral clustering initialized with the thresholded eigenvector of the graph Laplacian is already sufficient to achieve a significant improvement in terms of Cheeger cut values. Note that we are significantly faster than p-spectral clustering and in the same order of magnitude as the standard spectral relaxation. Finally, further improvement in terms of Cheeger cut values can be obtained by adding additional random initializations, however at the cost of an increased runtime.

7.7.3 Symmetric vertex expansion

In this section we evaluate our method for the symmetric vertex expansion problem. We again use a subset of the datasets obtained from the graph partitioning archive already used in the previous section.

As before, we compare against standard spectral clustering, where we threshold the second eigenvectors of the unnormalized and normalized graph Laplacian according to the VE criterion and take the best result. Similarly, we take the best result of thresholding the second eigenvector of the normalized and unnormalized graph p-Laplacian according to the VE criterion. We then repeat the experiment for the NVE criterion. For both criteria, in the case of the nonlinear IPM, we use 10 runs with random initializations.

The results can be seen in Table 7.5. One observes that for both VE and NVE criterion, our tight relaxation outperforms the other two methods in terms of obtained vertex expansion, however at the cost of a larger runtime. Moreover, currently it does not scale to large-scale graphs.

Note that here we used the general bundle method from Section 6.5 to solve the inner problem appearing in the nonlinear IPM. The advantage of this approach is that we are using a black-box algorithm, i.e. no knowledge about the special structure of the inner problem is required. In particular, the explicit form of the Lovász extension does not need to be known, as all computations can be done by only evaluating the original set functions. Of course, this means that we cannot expect to achieve optimal performance (in terms of runtime) when applying the general-purpose method. In contrast to that, for the Cheeger cut problem we explicitly exploited the structure of the inner problem. As a result, the corresponding runtimes are several orders of magnitude smaller for the Cheeger cut problem, see Fig. 7.4.

In conclusion, this experiment illustrates that the general purpose method from Section 6.5 can be used to solve the inner problem and leads to excellent results in terms of objective value. However, if additional knowledge of the structure of the inner problem is available, this should be exploited to achieve better performance in terms of runtime.

Table 7.5: Obtained VE and NVE values as well as total runtime on several datasets from the graph partitioning archive. In the case of the nonlinear IPM we perform 10 random initializations. Our method consistently outperforms the other methods in terms of obtained vertex expansion values, at the cost of a significantly larger runtime.

	Standard Spectral		p-Spectral ($p = 1.1$)		nonlinear IPM	
	VE	Time	VE	Time	VE	Time
add20	0.1925	3	0.2083	360	**0.1273**	3130
data	**0.04969**	4	**0.04969**	455	**0.04969**	2344
3elt	0.04292	5	0.03898	964	**0.03844**	6110
whitaker3	0.02723	12	0.02674	1063	**0.02673**	15733
cs4	0.06277	8	0.06271	4994	**0.05807**	37131

	Standard Spectral		p-Spectral ($p = 1.1$)		nonlinear IPM	
	NVE	Time	NVE	Time	NVE	Time
add20	0.04	3	0.05018	360	**0.01969**	3223
data	**0.004505**	4	**0.004505**	455	**0.004505**	3237
3elt	0.007346	5	0.006733	964	**0.006505**	5502
whitaker3	0.004603	12	0.00452	1063	**0.004476**	15569
cs4	0.01624	8	0.01617	4994	**0.01522**	58605

7.7.4 USPS and MNIST

We perform clustering on the full USPS and MNIST datasets of handwritten digits ($n = 9298$ and $n = 70000$), using the RCut as multi-cut criterion. We compare the performance of recursive standard spectral clustering, p-spectral clustering for $p = 1.1$, the TV-based method by Szlam and Bresson [2010], as well as the tight relaxations of the RCC and RCut criteria proposed in this chapter. For all methods, we use the recursive multi-partition scheme discussed in Section 7.6. As in the previous experiment, we perform one run initialized with the thresholded second eigenvector of the unnormalized graph Laplacian in the case of our method, and with the second eigenvector of the normalized graph Laplacian in the case of the method of Szlam and Bresson [2010]. In both cases we add 100 runs with random initialization.

In addition to the above recursive methods, we compare to the second variant of standard spectral clustering using k-means, as well as the TV-based multi-cut clustering method by Bresson et al. [2013] discussed in Section 7.6. Note that their code often returns a partition into less than 10 clusters. Thus we repeat their method 100 times (note that their default parameter is 30) and report the 10-partition with lowest RCut objective.

Table 7.6 shows the obtained RCut values and errors for all investigated methods. In the first two lines we compare the two versions of standard spectral clustering. The second variant produces slightly better cuts, but

for both datasets we obtain a worse error. Both variants of standard spectral clustering are outperformed by a large margin by the other methods, both in terms of RCut and error. Best results are obtained by the tight 1-spectral relaxation of the RCut criterion. Note that while the 1-spectral relaxations derived in this chapter are tight for the bi-partition version of the RCut and RCC criteria, this is not the case for the corresponding multi-cut criteria. However, in practice we observe a strong performance by the nonlinear IPM applied to the tight relaxation of the RCut objective. Slightly worse results are obtained by the 1-spectral relaxation of the RCC as well as the method by Szlam and Bresson [2010]. We also observe that all methods achieve a lower error compared to the standard spectral relaxation.

Table 7.6: Results of clustering for USPS and MNIST with $k = 10$ using the RCut multi-partition criterion (see Section 7.6).

	USPS		MNIST	
Method	RCut	Error	RCut	Error
Standard spectral (recursive)	0.8180	0.1686	0.2252	0.1883
Standard spectral (k-means)	0.7383	0.2088	0.2137	0.2650
Bresson et al. [2013]	0.6876	0.1366	0.1543	0.1257
p-spectral ($p = 1.1$)	0.6676	0.1308	0.1529	0.1293
Szlam and Bresson [2010]	0.6663	0.1309	0.1545	0.1318
1-spectral (Tight RCC)	0.6661	0.1349	0.1507	0.1244
1-spectral (Tight RCut)	**0.6629**	**0.1301**	**0.1499**	**0.1236**

Note that the error has been computed by computing the disagreement of the dominant class label in each cluster with the true label of each point. In Table 7.7 we provide the confusion matrix for MNIST for the tight relaxation of the RCC criterion. Here, each column corresponds to one cluster and we count for each cluster the number of appearances of each digit (first column). The first row gives the dominant class label (i.e. digit) of each cluster. We observe that the true cluster corresponding to 1 has been split into two clusters, and the clusters corresponding to 4 and 9 have been merged into one cluster with dominant label 4. Thus there is no cluster with dominant label 9. Apart from that, the class separation is quite good, as the remaining digits have been grouped into separate clusters almost perfectly.

For comparison, we give the confusion matrix for the variant of standard spectral clustering using higher eigenvectors. One observes that the clustering corresponds much less to the true class structure of the problem, as the digits 1 have been split into three clusters, and 3,5 and 8 have been merged into one cluster. Moreover, the majority of the points with label 9 appear in the clusters corresponding to digits 4 and 7.

Table 7.7: Confusion table for MNIST for the tight relaxation of RCC (*top*) and standard spectral clustering (*bottom*). Here, each column corresponds to a cluster while each row corresponds to a true label. The first row denotes the dominant labels in each cluster. In the case of 1-Spectral clustering, one class has been split and two classes have been merged. In contrast to that, in standard spectral clustering, three classes have been split and seven classes have been merged.

	0	**1**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	**6846**	4	4	7	0	5	7	25	4	1
1	1	**4278**	**3525**	23	0	25	0	7	16	2
2	49	31	22	**6690**	21	14	3	13	125	22
3	2	8	5	37	**6880**	40	79	1	56	33
4	3	16	28	1	1	**6743**	0	20	9	3
5	15	1	2	2	50	58	**6104**	60	7	14
6	20	5	9	0	1	8	30	**6798**	0	5
7	1	34	43	23	1	102	0	0	**7089**	0
8	15	45	34	14	90	120	120	24	23	**6340**
9	17	9	5	4	103	**6706**	12	4	84	14

	0	**1**	**1**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	**6792**	0	0	4	4	59	6	15	19	4
1	0	**3171**	**2660**	**1946**	29	28	24	0	1	18
2	46	34	14	12	**6552**	133	25	13	13	148
3	1	4	4	2	28	**6957**	31	58	1	55
4	3	21	16	27	1	4	**6695**	2	20	35
5	15	2	1	1	1	**3112**	44	**3089**	36	12
6	23	9	3	7	1	45	9	251	**6528**	0
7	2	37	18	32	16	4	119	2	0	**7063**
8	16	33	26	20	10	**6150**	92	397	12	69
9	15	3	6	5	5	158	**4776**	2	4	**1984**

Chapter 8

Constrained local clustering

Recently, there has been a strong interest in local methods for clustering. The previous work can be divided into two categories. In the seminal article by Spielman and Teng [2004], an algorithm was developed that finds a subset near a given seed vertex with small normalized cut or normalized Cheeger cut value, with running time linear in the size of the obtained cluster. The proposed algorithm and subsequent work [Andersen and Lang, 2006, Andersen et al., 2006, Andersen and Peres, 2009, Oveis Gharan and Trevisan, 2012, Zhu et al., 2013] use random walks to explore the graph locally, without considering the whole graph.

In the second line of work, the runtime requirement is dropped and the task is formulated as explicit optimization problem [Mahoney et al., 2012, Maji et al., 2011, Hansen and Mahoney, 2012]. The goal is to find the optimal normalized cut subject to a seed constraint and an upper bound on the volume of the set. We now use the results from Chapter 4 to derive a method for the above problem aligned with the second type of approaches. In contrast to previous work, our method will guarantee that all constraints are fulfilled by the solution.

8.1 The constrained local clustering problem

In this section we introduce the local clustering problem. We are in the same graph-based setting as in the last chapter, see Section 7.1. Let J denote the set of seed vertices, S a symmetric balancing function (e.g. $S(C) = \operatorname{vol}_d(C)\operatorname{vol}_d(\overline{C})$ for the normalized cut) and let $\operatorname{vol}_h(C)$ be the general volume of set C, where $h \in \mathbb{R}^n_+$ are vertex weights. The general local clustering problem can then be formulated as

$$\begin{aligned} &\min_{C \subset V} \frac{\operatorname{cut}(C, \overline{C})}{\widehat{S}(C)} \\ &\text{subject to}: \ \operatorname{vol}_h(C) \leq k, \ \text{ and } \ J \subset C. \end{aligned} \tag{8.1}$$

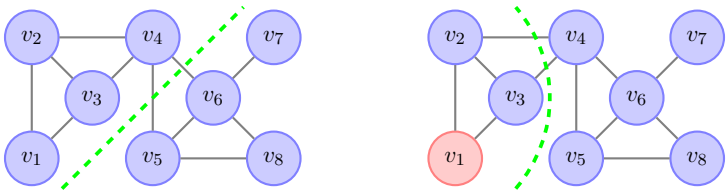

Figure 8.1: *Left:* Optimal RCut (green). *Right:* Optimal RCut with subset (red) and constraint $|C| \leq 3$.

Fig. 8.1 gives an example of local clustering. In the left example we optimize RCut without any additional constraints, leading to a cluster with $\text{cut}(C, \overline{C}) = 2$ and $|C| = \left|\overline{C}\right| = 4$, and hence $\text{RCut}(C, \overline{C}) = 1$. On the right side we add the additional constraint $|C| \leq 3$, as well as a subset constraint. Since the clustering one the left does not satisfy the constraint $|C| \leq 3$, we obtain a different clustering with slightly higher objective value ($\text{RCut}(C, \overline{C}) = \frac{16}{15}$) which satisfies subset and size constraint.

Several authors considered a variant of the above problem. Starting with the work of Spielman and Teng [2004], initially, the goal was to develop an *algorithm* to find a subset near a given seed vertex with small NCut or NCC value in running time nearly linear in the size of the obtained cluster. Their algorithm performs a lazy random walk with the transition matrix $M = \frac{1}{2}\left(I + WD^{-1}\right)$, where D is the degree matrix of the graph and the initial distribution is concentrated on the seed vertex. Given a cut with NCC less than ϕ, they guarantee that after a specified number of steps, optimal thresholding of the random walk vector will yield a set C having a normalized Cheeger cut value in $O(\phi^{\frac{1}{3}} \log^{O(1)} |V|)$.

Andersen et al. [2006] showed that an improved bound of $O(\sqrt{\phi\,|E|})$ can be achieved by using PageRank vectors. Further improvements were done by Spielman and Teng [2013], Andersen and Peres [2009], Oveis Gharan and Trevisan [2012]. Recently, Zhu et al. [2013] studied a variant of the PageRank-based algorithm, where they also relate the performance of the algorithm to the connectedness of the set, using a definition based on the minimum NCC inside a cluster. They then obtain better bounds on the obtained NCC under the condition that the cluster is well-connected.

The lazy random walk was applied for community detection in networks by Andersen and Lang [2006]. Moreover, it was used to construct spectral sparsifiers of graphs in nearly linear time [Spielman and Teng, 2011] and near linear time algorithms to solve linear systems [Spielman and Teng, 2006].

Note that in the above type of approaches, the focus was on obtaining an efficient algorithm to obtain a good partition around the given seed in nearly linear time. Thus typically the volume and seed constraints were not formulated as hard constraints required to be fulfilled by the solution. In

contrast to these greedy approaches, Mahoney et al. [2012] give up the runtime requirement and formulate the task as an explicit *optimization problem*, with the goal to find the optimal normalized cut subject to a seed constraint and an upper bound on the volume of the set containing the seed set.

Motivated by the standard spectral relaxation of the normalized cut problem, they derive a spectral-type relaxation which is biased towards solutions fulfilling the seed constraint. The resulting problem is then transformed into an equivalent SDP which can be solved globally optimal. Their method has been applied in semi-supervised image segmentation [Maji et al., 2011] and for community detection around a query set [Mahoney et al., 2012].

The continuous solution is transformed into a set via optimal thresholding. Since this is not guaranteed to yield a feasible set, i.e. one that satisfies both constraints, Mahoney et al. [2012] suggest to perform *constrained* optimal thresholding, i.e. consider only thresholds that yield feasible sets. However, this comes at the cost of losing the derived approximation guarantees. In a recent generalization of their work, Hansen and Mahoney [2012] compute a sequence of locally-biased eigenvectors, the first of which corresponds to the solution of the spectral-type relaxation of Mahoney et al. [2012].

The problem (8.1) is an extended version of the problem considered by Mahoney et al. [2012]. The choice of the balancing function S allows the user to influence the trade-off between getting a partition with small cut and a balanced partition. One could also add more complex constraints such as an upper bound on the diameter of C, or must- and cannot-link constraints as done by Rangapuram and Hein [2012].

8.2 Tight relaxation of constrained NCut problem

In the following, we will derive a tight relaxation of the local clustering problem. In order to compare to the method of Mahoney et al. [2012] in the experiments, we restrict ourselves to the normalized cut with volume constraints, i.e. $\widehat{S}(C) = \operatorname{vol}_d(C)\operatorname{vol}_d(\overline{C})$, however the derivation can be done similarly for other balancing terms. The NCut problem with subset constraints and general volume constraints is then for a subset J given as

$$\begin{aligned} &\min_{C \subset V} \frac{\operatorname{cut}(C, \overline{C})}{\operatorname{vol}(C)\operatorname{vol}(\overline{C})} \\ &\text{subject to}: \ \operatorname{vol}_h(C) \leq k, \ \text{ and } \ J \subset C. \end{aligned} \tag{8.2}$$

8.2.1 Elimination of volume constraints

First, we integrate the volume constraint via a penalty term. Adding the constraint $\operatorname{vol}_h(C) \leq k$ for some nonnegative function $h : V \to \mathbb{R}_+$ leads to the penalty term $\widehat{T}(C) = \max\{\operatorname{vol}_h(C) - k, 0\}$.

Proposition 8.1. *The penalty term $\widehat{T}(C)$ is equal to $\widehat{T}(C) = \mathrm{vol}_h(C) - \min\{k, \mathrm{vol}(C)\}$, which is a difference of submodular functions. Its Lovász extension is a difference of convex functions.*

Proof. By Prop. 2.22, the function $\mathrm{vol}_h(C)$ is modular. Moreover, it can be easily derived from the definition of submodularity that the pointwise minimum of a constant and an increasing submodular function is submodular, which implies the submodularity of $\min\{k, \mathrm{vol}(C)\}$. By Prop. 2.15, the Lovász extension of $\widehat{T}$ is the difference of the Lovász extensions of $\mathrm{vol}_h(C)$ and $\min\{k, \mathrm{vol}(C)\}$, which are both convex by Prop. 2.19. □

We now can use the penalty term $\widehat{T}(C)$ to transform the constrained fractional set program into an equivalent unconstrained set program.

Lemma 8.2. *The problem* (8.2) *is equivalent to the problem*

$$\begin{aligned} \min_{C \subset V} \ & \frac{\mathrm{cut}(C, \overline{C}) + \gamma \widehat{T}(C)}{\mathrm{vol}(C)\,\mathrm{vol}(\overline{C})} \\ \text{subject to :} \ & J \subset C. \end{aligned} \tag{8.3}$$

where $\gamma > \frac{\mathrm{cut}(C_0, \overline{C_0})\,\mathrm{vol}(V)^2}{4\,\theta\,\mathrm{vol}(C_0)\,\mathrm{vol}(\overline{C_0})}$ for some feasible set $C_0 \subset V$.

Proof. This follows directly from Lemma 4.5. For the bound on γ one uses the fact that $\mathrm{vol}(C)\ \mathrm{vol}(\overline{C})$ is maximal if $\mathrm{vol}(C) = \mathrm{vol}(\overline{C}) = \frac{1}{2}\,\mathrm{vol}(V)$. □

8.2.2 Direct integration of seed constraint

One way to handle the seed constraint $J \subset C$ is to rewrite it as inequality constraint $|J \cap C| - |J| \geq 0$ and add a similar penalty function to the numerator of (8.3), see Section 8.2.3. However, using the structure of the problem, a more direct way is possible. The idea is to use that $C = A \cup J$ for some set $A \subset V$ with $A \cap J = \emptyset$, thus the objective can be reformulated in terms of the set A. This reduces the problem to an optimization problem on the graph with vertices $V' = V \backslash J$, as we will show below.

Lemma 8.3. *The problem* (8.3) *is equivalent to the problem*

$$\min_{A \subset V'} \ \frac{\mathrm{cut}(A, V' \backslash A) + \mathrm{cut}(J, V') - \mathrm{cut}(J, A) + \gamma \widehat{T}_{k'}(A)}{\mathrm{vol}_d(A)\,\mathrm{vol}_d(V' \backslash A) + \mathrm{vol}_d(J)\,\mathrm{vol}_d(V') - \mathrm{vol}_d(J)\,\mathrm{vol}_d(A)}, \tag{8.4}$$

where $k' = k - \mathrm{vol}_h(J)$ and

$$\gamma > \frac{\mathrm{cut}(A_0, V' \backslash A_0) + \mathrm{cut}(J, V') - \mathrm{cut}(J, A_0))\,\mathrm{vol}_d(V)^2}{4\,\theta(\mathrm{vol}_d(A_0)\,\mathrm{vol}_d(V' \backslash A_0) + \mathrm{vol}_d(J)\,\mathrm{vol}_d(V' \backslash A_0)))}$$

for some feasible set $A_0 \subset V'$. Solutions C^* of (8.3) *and A^* of* (8.4) *are related via $C^* = A^* \cup J$.

Proof. Writing $C = A \cup J$, where $A \subset V$ with $A \cap J = \emptyset$, the individual terms in (8.3) can be decomposed as follows:

$$\begin{aligned}\operatorname{cut}(C, \overline{C}) &= \sum_{i \in C, j \in V \setminus C} w_{ij} = \sum_{i \in A, j \in V \setminus (A \cup J)} w_{ij} + \sum_{i \in J, j \in V \setminus (A \cup J)} w_{ij} \\ &= \sum_{i \in A, j \in V' \setminus A} w_{ij} + \sum_{i \in J, j \in V'} w_{ij} - \sum_{i \in J, j \in A} w_{ij} \\ &= \operatorname{cut}(A, V' \setminus A) + \operatorname{cut}(J, V') - \operatorname{cut}(J, A),\end{aligned}$$

$$\begin{aligned}\operatorname{vol}_d(C) \operatorname{vol}_d(\overline{C}) &= \sum_{i \in C} d_i \sum_{j \in V \setminus C} d_j = \sum_{i \in A} d_i \sum_{j \in V \setminus (A \cup J)} d_j + \sum_{i \in J} d_i \sum_{j \in V \setminus (A \cup J)} d_j \\ &= \sum_{i \in A} d_i \sum_{j \in V' \setminus A} d_j + \sum_{i \in J} d_i \sum_{j \in V' \setminus A} d_j \\ &= \operatorname{vol}_d(A) \operatorname{vol}_d(V' \setminus A) + \operatorname{vol}_d(J) \operatorname{vol}_d(V' \setminus A), \quad \text{and}\end{aligned}$$

$$\begin{aligned}\widehat{T}_k(C) &= \max\{\operatorname{vol}_h(C) - k, 0\} = \max\{\operatorname{vol}_h(A) - (k - \operatorname{vol}_h(J)), 0\} \\ &= \widehat{T}_{k'}(A), \quad \text{where } k' = k - \operatorname{vol}_h(J).\end{aligned}$$

Replacing the terms in (8.3) gives the result. Thus a solution C^* of (8.3) is found by computing a solution A^* of (8.4) and setting $C^* = A^* \cup J$. □

The relation between subsets A, J and C is illustrated in Fig. 8.2. Note that one is now working on the reduced graph with vertices $V' = V \setminus J$. Thus, the set $V' \setminus A$ is just the complement of A on the reduced graph V'. However, we will use the explicit notation $V' \setminus A$ in this section to avoid confusion with the complement on the original graph, $V \setminus A$. Moreover, note that the terms $\operatorname{vol}_d(A)$ appearing in the above lemma still use the degree d of the original graph, i.e. $\operatorname{vol}_d(A) = \sum_{i \in V} d_i$, where $d_i = \sum_{j \in V} w_{ij}$ denotes the degree of vertex i on the original graph with vertex set V.

In order to derive the tight relaxation via Theorem 4.2, we will use the Lovász extensions of the set functions in (8.4). While the above ratio has a slightly more complicated structure that the original one in (8.3), we will see that the additional terms $\operatorname{cut}(J, A)$ and $\operatorname{vol}_d(J) \operatorname{vol}_d(A)$ are both modular and thus have a linear Lovász extension. Thus, the corresponding terms in the inner problem of RatioDCA can be easily handled in the optimization. Moreover, $\operatorname{cut}(J, V')$ and $\operatorname{vol}_d(J) \operatorname{vol}_d(V')$ are constants.

Note that the Lovász extension of a set function $\widehat{S}$ in Def. 2.12 requires the function to satisfy $\widehat{S}(\emptyset) = 0$. Since this requirement is not fulfilled for constant functions, we replace the constant set functions $\operatorname{vol}_d(J) \operatorname{vol}_d(V \setminus J)$ and $\operatorname{cut}(J, V \setminus J)$ by $\operatorname{vol}_d(J) \operatorname{vol}_d(V \setminus J) \widehat{P}(A)$ and $\operatorname{cut}(J, V \setminus J) \widehat{P}(A)$, respectively, where $\widehat{P}$ is defined as $\widehat{P}(A) = 1$ for $A \neq \emptyset$ and $\widehat{P}(\emptyset) = 0$. This leads

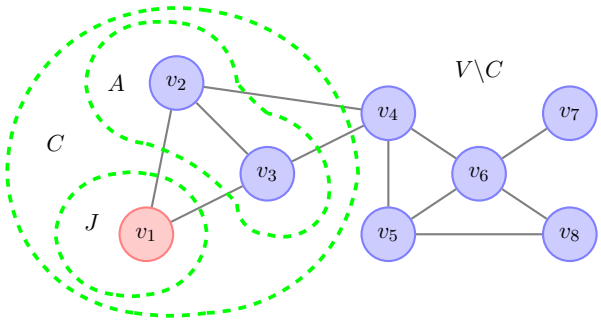

Figure 8.2: Relation between subsets A, J and C used in Lemma 8.3. After integrating the seed subset, we work on the graph consisting of the blue vertices instead of the whole graph.

to the problem

$$\min_{A \subset V'} \frac{\mathrm{cut}(A, V' \backslash A) + \mathrm{cut}(J, V')\widehat{P}(A) - \mathrm{cut}(J, A) + \gamma \widehat{T}_{k'}(A)}{\mathrm{vol}_d(A)\,\mathrm{vol}_d(V' \backslash A) + \mathrm{vol}_d(J)\,\mathrm{vol}_d(V')\widehat{P}(A) - \mathrm{vol}_d(J)\,\mathrm{vol}_d(A)}. \quad (8.5)$$

The only difference to (8.4) lies in the treatment of the empty set. Note that with $\frac{0}{0} := \infty$ the empty set can not be optimal for problem (8.5). Given an optimal solution A^* of (8.5), one then considers either $A^* \cup J$ or J, depending on whichever has lower objective, which then implies equivalence to (8.5).

The resulting tight relaxation will be a minimization problem over $\mathbb{R}^m$ with $m = |V'| = |V \backslash J|$ and we assume wlog that the first m vertices of V are the ones in V'. Moreover, we use the notation $f_{\max} = \max_{i=1,\dots,m} f_i$ for $f \in \mathbb{R}^m$, and $d_i^{(A)} = \sum_{j \in A} w_{ij}$. In the following we will derive some Lovász extensions of the involved set functions.

Lemma 8.4. *The set function $c\widehat{P}(A)$ is submodular for every $c \in \mathbb{R}^+$. Its Lovász extension is given by cf_{max}.*

Proof. The Lovász extension follows directly from the second definition of the Lovász extension in Def. 2.12, noting that all the terms in the first sum are zero. Since the Lovász extension is a convex function for all $c \geq 0$, the function $c\widehat{P}(A)$ is submodular by Prop. 2.19. □

Lemma 8.5. *The set function $\mathrm{cut}(J, A)$ for a fixed J is modular. Its Lovász extension is given by $\langle d^{(J)}, f \rangle$.*

Proof. We can rewrite the set function as

$$\mathrm{cut}(J, A) = \sum_{i \in A, j \in J} w_{ij} = \sum_{i \in A} d_i^{(J)},$$

were $d_i^{(J)} = \sum_{j \in J} w_{ij}$ as defined above. Thus we can interpret the set function as generalized volume function, i.e. $\mathrm{cut}(J, A) = \mathrm{vol}_{d^{(J)}}(A)$. The results then follow directly from Prop. 2.22. □

As shown in Lemma 8.1, the penalty term $\widehat{T}_{k'}(A)$ can be written as $\widehat{T}_{k'}(A) = \mathrm{vol}_h(A) - \min\{k', \mathrm{vol}_h(A)\}$. For the sake of brevity, we do not specify the Lovász extension of the set function $\widehat{T}^{(2)}_{k'}(A) = \min\{k', \mathrm{vol}_h(A)\}$ in a closed form. Recall from Section 5 that in the implementation in RatioDCA we will need only an element of the subdifferential of the Lovász extension of $\widehat{T}^{(2)}_{k'}(A)$ which we will present in the next lemma.

Lemma 8.6. *An element $t^{(2)}_{k'}$ of the subdifferential of the Lovász extension $T^{(2)}_{k'}(f)$ of $\widehat{T}^{(2)}_{k'}(A) = \min\{k', \mathrm{vol}_h(A)\}$ is given by*

$$\big(t^{(2)}_{k'}(f)\big)_{j_i} = \begin{cases} 0 & \mathrm{vol}_h(A_{i+1}) > k' \\ k' - \mathrm{vol}_h(A_{i+1}) & \mathrm{vol}_h(A_i) \geq k', \\ & \quad \mathrm{vol}_h(A_{i+1}) \leq k' \\ h_{j_i} & \mathrm{vol}_h(A_i) < k' \end{cases},$$

where j_i denotes the index of the i-th smallest component of the vector f. Moreover, we have $T^{(2)}_{k'}(f) = \big\langle f, t^{(2)}_{k'}(f) \big\rangle$ for all $f \in \mathbb{R}^n$.

Proof. This directly follows from Lemma 2.20, using the fact the sequence $\mathrm{vol}_h(A_i)$ is monotonically decreasing in i. □

The above Lovász extensions lead to the following tight relaxation of (8.5):

Theorem 8.7 (Tight relaxation of constrained normalized cut). *The problem in* (8.5) *is equivalent to the problem*

$$\min_{f \in \mathbb{R}^m_+} \frac{R_1(f) - R_2(f)}{S_1(f) - S_2(f)}, \tag{8.6}$$

where the convex functions R_1, R_2, S_1, S_2 are given as

$$\begin{aligned} R_1(f) &= \tfrac{1}{2} \textstyle\sum_{i,j \in V'} w_{ij} \big| f_i - f_j \big| + \mathrm{cut}(J, V')\, f_{\max} + \gamma \, \big\langle (h_i)_{i=1}^m, f \big\rangle, \\ R_2(f) &= \big\langle \gamma \, t^{(2)}_{k'}(f) + (d_i^{(J)})_{i=1}^m, f \big\rangle, \\ S_1(f) &= \tfrac{1}{2} \mathrm{vol}_d(V') \textstyle\sum_{i \in V'} d_i \big| f_i - \mathrm{mean}_d(f) \big| + \mathrm{vol}_d(J)\, \mathrm{vol}_d(V')\, f_{\max}, \\ S_2(f) &= \mathrm{vol}_d(J) \big\langle (d_i)_{i=1}^m, f \big\rangle. \end{aligned}$$

Proof. The objective in (8.5) can be written as a ratio of two differences of submodular set functions as follows

$$\min_{A \subset V'} \frac{\widehat{R}_1(A) - \widehat{R}_2(A)}{\widehat{S}_1(A) - \widehat{S}_2(A)},$$

where the submodular functions $\widehat{R}_1$, $\widehat{R}_2$, $\widehat{S}_1$, $\widehat{S}_2$ are given as

$$\begin{aligned}
\widehat{R}_1(A) &= \operatorname{cut}(A, V'\backslash A) + \operatorname{cut}(J, V')\widehat{P}(A) + \gamma \operatorname{vol}_h(A), \\
\widehat{R}_2(A) &= \gamma \min\{k', \operatorname{vol}_h(A)\} + \operatorname{cut}(J, A), \\
\widehat{S}_1(A) &= \operatorname{vol}_d(A) \operatorname{vol}_d(V'\backslash A) + \operatorname{vol}_d(J) \operatorname{vol}_d(V')\widehat{P}(A), \quad \text{and} \\
\widehat{S}_2(A) &= \operatorname{vol}_d(J) \operatorname{vol}_d(A).
\end{aligned}$$

We have used Prop. 8.1 for the decomposition of $\widehat{T}(A)$ into a difference of submodular functions, and Lemmas 8.4 and 8.5 for the submodularity of $\widehat{P}$ and $\operatorname{cut}(J, A)$. The submodularity of $\operatorname{vol}_d(A)$, $\operatorname{cut}(A, V'\backslash A)$ as well as $\operatorname{vol}_d(A) \operatorname{vol}_d(V'\backslash A)$ is shown in Prop. 2.22, Prop. 2.23 and Lemma 7.24.

One now replaces the set functions by their Lovász extensions, derived in Prop. 2.22 and Prop. 2.23 for $\operatorname{vol}_h(A)$ and $\operatorname{cut}(A, V'\backslash A)$, Lemma 8.4 and Lemma 8.5 for $\widehat{P}(A)$ and $\operatorname{cut}(J, A)$ as well as Lemma 8.6 for $\min\{k', \operatorname{vol}_h(A)\}$. For the set function $\operatorname{vol}_d(A) \operatorname{vol}_d(V'\backslash A)$ we use the continuous extension from Lemma 7.24. The result then follows by Theorem 4.4. □

8.2.3 Seed constraint via penalty function

Here we discuss the alternative approach where the subset constraint is incorporated directly into the objective. We are now working again on the original graph with vertex set V, and the resulting tight relaxation will be an optimization problem over $\mathbb{R}^n_+$.

Lemma 8.8. *The problem* (8.2) *is equivalent to the problem*

$$\min_{C \subset V} \frac{\operatorname{cut}(C, \overline{C}) + \gamma_1 \widehat{T}_1(C) + \gamma_2 \widehat{T}_2(C)}{\operatorname{vol}(C) \operatorname{vol}(\overline{C})} \tag{8.7}$$

where $\widehat{T}_1(C) = |J| - |C \cap J|$ *and* $\widehat{T}_2(C) = \operatorname{vol}_h(C) - \min\{k, \operatorname{vol}(C)\}$ *and* $\gamma_1, \gamma_2 > \frac{\operatorname{cut}(C_0, \overline{C_0}) \operatorname{vol}(V)^2}{4\,\theta \operatorname{vol}(C_0) \operatorname{vol}(\overline{C_0})}$ *for some feasible set* $C_0 \subset V$.

Proof. The subset constraint can be rewritten as $|C \cap J| = |J|$. Writing the equality as two inequalities, one can apply Lemma 4.5, which yields an equivalent problem using the penalty terms

$$\max\{|C \cap J| - |J|, 0\} \quad \text{and} \quad \max\{|J| - |C \cap J|, 0\} \tag{8.8}$$

for the subset constraint, and $\operatorname{vol}_h(C) - \min\{k, \operatorname{vol}(C)\}$ for the volume constraint, see Section 8.2.1. Using the fact that $|J| \geq |C \cap J|$, one can conclude that the first term in (8.8) is always zero, while the second one is equal to $|J| - |C \cap J|$. Thus one obtains the problem in (8.7). □

As before we replace the constant $|J|$ by the function $|J|\,\widehat{P}(C)$, see the discussion after Lemma 8.3. This leads to the problem

$$\min_{C\subset V} \frac{\operatorname{cut}(C,\overline{C}) + \gamma_1 \big(|J|\,\widehat{P}(C) - |C\cap J| \big) + \gamma_2\,\widehat{T}_2(C)}{\operatorname{vol}(C)\operatorname{vol}(\overline{C})}. \tag{8.9}$$

The following lemma states the Lovász extension of the function $|C \cap J|$.

Lemma 8.9. *The set function $|C \cap J|$ is a modular function. Its Lovász extension is given by $\langle \mathbf{1}_J, f\rangle$, where $\mathbf{1}_J$ is the indicator vector of the set J.*

Proof. We can rewrite the set function as $|C\cap J| = \sum_{i\in C, j\in J} 1 = \sum_{i\in C} (\mathbf{1}_J)_i = \operatorname{vol}_{\mathbf{1}_J}(C)$. The results then follow directly from Prop. 2.22. □

We now present a tight relaxation of the problem in (8.9).

Theorem 8.10 (Tight relaxation with subset as penalty). *The problem in* (8.9) *is equivalent to the problem*

$$\min_{f\in\mathbb{R}^n_+} \frac{R_1(f) - R_2(f)}{S(f)}, \tag{8.10}$$

where the convex functions R_1, R_2 and S are given as

$$\begin{aligned}
R_1(f) &= \tfrac{1}{2}\textstyle\sum_{i,j\in V} w_{ij}\big|f_i - f_j\big| + \gamma_1\,|J|\;f_{\max} + \gamma_2\,\big\langle h, f\big\rangle,\\
R_2(f) &= \gamma_1\big\langle \mathbf{1}_J, f\big\rangle + \gamma_2\big\langle t_k^{(2)}(f), f\big\rangle \quad \textit{and}\\
S(f) &= \tfrac{1}{2}\operatorname{vol}(V)\textstyle\sum_{i\in V} d_i\,|f_i - \operatorname{mean}_d(f)|\,.
\end{aligned}$$

Proof. The objective in (8.9) can be written as a ratio of a difference of submodular set functions and a submodular set function as follows

$$\min_{C\subset V} \frac{\widehat{R}_1(C) - \widehat{R}_2(C)}{\widehat{S}(C)},$$

where the submodular functions $\widehat{R}_1$, $\widehat{R}_2$ and $\widehat{S}$ are given as

$$\begin{aligned}
\widehat{R}_1(C) &= \operatorname{cut}(C,\overline{C}) + \gamma_1\,|J|\,\widehat{P}(C) + \gamma_2\operatorname{vol}_h(C),\\
\widehat{R}_2(C) &= \gamma_1\,|J\cap C| + \gamma_2\min\{k, \operatorname{vol}_h(C)\} \quad \text{and}\\
\widehat{S}(C) &= \operatorname{vol}(C)\operatorname{vol}(\overline{C}),
\end{aligned}$$

We have used Prop. 8.1 for the submodular decomposition of $\widehat{T}(A)$, Lemmas 8.4 and 8.9 for the submodularity of $\widehat{P}(C)$ and $|C\cap J|$, as well as Prop. 2.23 and Lemma 7.24 for the submodularity of $\operatorname{cut}(C,\overline{C})$ and $\operatorname{vol}(C)\operatorname{vol}(\overline{C})$.

The functions R_1, R_2 and S are then obtained by replacing the set functions by their Lovász extensions derived in Prop. 2.22 and 2.23, Lemma 8.4, 8.6 and Lemma 8.9 as well as the continuous extension derived in Lemma 7.24. The result then follows by Theorem 4.4. □

While the main focus of this work was to find solutions that satisfy both seed and volume constraints (which is guaranteed if γ satisfies the bound given above), the penalty approach can also be useful in a slightly different setting: Assume that we have noisy seed constraints, or we are in a setting where we are willing to trade some violated constraints for a much better objective value. Incorporating these noisy or weak constraints into the problem via a penalty approach, we can use the choice of γ to control the trade-off between satisfying all constraints and achieving an optimal objective value. We will discuss an example in the experiments in Section 8.4.2.

8.3 Solution via RatioDCA

As both numerator and denominator of the tight relaxations (8.6) and (8.10) are 1-homogeneous d.c. functions, we can apply the RatioDCA of Section 5. Recall the general form of the inner problem,

$$\min_{\substack{u\in\mathbb{R}^n_+,\\ \|u\|_2\leq 1}} \Big\{ R_1(u) - \big\langle u, r_2(f^k)\big\rangle + \lambda^k\Big(S_2(u) - \big\langle u, s_1(f^k)\big\rangle\Big)\Big\},$$

where $r_2(f^k) \in \partial R_2(f^k)$ and $s_1(f^k) \in \partial S_1(f^k)$. We now derive subgradients for the specific problems in (8.6) and (8.10).

In the case of a linear function $\langle g, f\rangle$, the subgradient is given by g. Moreover, one easily checks that an element of the subgradient of $f_{\max}$ is given as $\frac{1}{|C_{\max f}|}\mathbf{1}_{C_{\max f}}$, where $C_{\max f} = \{i \in V \mid f_i = f_{\max}\}$ is the set of indices where f has its largest value. The following lemma gives a subgradient of the term $\sum_{i\in V} d_i\,|f_i - \mathrm{mean}_d(f)|$ appearing in S_1 for both problems.

Lemma 8.11. *The subdifferential of* $\sum_{i\in V} d_i\,|f_i - \mathrm{mean}_d(f)|$ *is given as*

$$\Big(D - \frac{dd^T}{\mathrm{vol}_d(V)}\Big)\,\mathrm{sign}\Big(f - \frac{\langle d, f\rangle}{\mathrm{vol}_d(V)}\mathbf{1}\Big), \quad \textit{where } \mathrm{sign}(x)_i = \begin{cases} -1, & x < 0,\\ [-1,1], & x = 0,\\ 1, & x > 0.\end{cases}$$

Proof. We can write

$$\sum_{i\in V}\Big|d_i\Big(f_i - \frac{\langle d, f\rangle}{\mathrm{vol}_d(V)}\Big)\Big| = \Big\|D\Big(f - \frac{\langle d, f\rangle}{\mathrm{vol}_d(V)}\mathbf{1}\Big)\Big\|_1 = \Big\|\Big(D - \frac{dd^T}{\mathrm{vol}_d(V)}\Big)f\Big\|_1.$$

Noting that $\partial\left(\|x\|_1\right)_k = \mathrm{sign}(x_k)$, we apply the chain rule for subdifferentials

(see e.g. Theorem 23.9 in Rockafellar [1970]) and obtain

$$\partial\Big(\sum_{i\in V} d_i\,|f_i - \mathrm{mean}_d(f)|\Big) = \Big(D - \frac{dd^T}{\mathrm{vol}_d(V)}\Big)^T \mathrm{sign}\Big(\big(D - \frac{dd^T}{\mathrm{vol}_d(V)}\big)f\Big)$$
$$= \Big(D - \frac{dd^T}{\mathrm{vol}_d(V)}\Big)\,\mathrm{sign}\Big(f - \frac{\langle d, f\rangle}{\mathrm{vol}_d(V)}\mathbf{1}\Big),$$

where we have used the symmetry of D and dd^T as well as the fact that $d_i > 0,\ \forall i \in V$. □

It turns out that the remaining terms $R_1(f) + \lambda^k S_2(f)$ have the same structure for (8.6) and (8.10). In both cases the inner problem has the form

$$\min_{\substack{f\in\mathbb{R}^m_+ \\ \|f\|_2\le 1}} \Big\{\tfrac{1}{2}\sum_{i,j}^{m} w_{ij}|f_i - f_j| + c_1 f_{\max} + \langle f, c_2^k\rangle\Big\}, \tag{8.11}$$

where $c_1 \in \mathbb{R}$ is a constant and c_2^k is a vector depending on the current iterate f^k. The explicit values are given for the problem in (8.6) as

$$\begin{aligned} c_1 \;&= \mathrm{cut}(J, V'), \\ c_2^k \;&= \gamma\,(h_i)_{i=1}^m \;-\; (d_i^{(J)})_{i=1}^m \;+\; \lambda^k \,\mathrm{vol}_d(J)(d_i)_{i=1}^m \\ &\quad -\; \gamma t_{k'}^{(2)}(f^k) \;-\; \lambda^k \tfrac{1}{2}\,\mathrm{vol}(V')v(f^k) \;-\; \lambda^k\,\mathrm{vol}_d(J)\,\mathrm{vol}_d(V')\mathbf{1}_{C_{\max f^k}}, \end{aligned} \tag{8.12}$$

where $v(f^k) \in \partial\left(\sum_{i\in V} d_i\,|f_i - \mathrm{mean}_d(f)|\right)$, and for the problem in (8.10) as

$$\begin{aligned} c_1 &= \gamma_1\,|J|\,, \\ c_2^k &= \gamma_2 h - \lambda^k \tfrac{1}{2}\,\mathrm{vol}(V)v(f^k) - \gamma_1\mathbf{1}_J - \gamma_2 t_k^{(2)}(f). \end{aligned} \tag{8.13}$$

Alg. 18 summarizes the algorithmic scheme for both cases.

18 Algorithm for constrained normalized cut problem

1: **Input:** weight matrix W
2: **Initialization:** nonconstant f^0 with $\left\|f^0\right\|_2 = 1$, accuracy ϵ
3: **repeat**
4: Compute c_1 and c^k according to (8.12) for (8.6) or (8.13) for (8.10).
5: $f^{k+1} = \underset{\substack{f\in\mathbb{R}^m_+ \,/\, f\in\mathbb{R}^n_+ \\ \|f\|_2\le 1}}{\arg\min} \Big\{\frac{1}{2}\sum_{i,j}^m w_{ij}|f_i - f_j| + c_1 f_{\max} + \langle f, c_2^k\rangle\Big\}$,
6: $\lambda^{k+1} = \frac{R_1(f^k) - R_2(f^k)}{S_1(f^k) - S_2(f^k)}$
7: **until** $\frac{\left|\lambda^{k+1} - \lambda^k\right|}{\lambda^k} < \epsilon$

Theorem 8.12 (Convergence). *Let Q be the functional in* (8.6) *or* (8.10)*, depending on the choice of c_1 and c^k. The sequence f^k produced by Alg. 18 satisfies $Q(f^k) > Q(f^{k+1})$ for all $k \geq 0$ or terminates. Moreover, f^k has a subsequence converging to a solution of the eigenproblem associated to Q.*

Proof. This is a direct corollary of Prop. 5.9 and Theorem 5.13. □

We omit the exact form of the eigenproblem associated to the problems in (8.6) or (8.10). Similar to the unconstrained case from Chapter 7, one can give the following cut improvement guarantee.

Theorem 8.13 (Cut improvement). *Let C be any feasible set, f denote the result of Alg. 18 after initializing with $\frac{1}{|C|}\mathbf{1}_C$, and C_f be the set obtained by optimal thresholding of f. Either Alg. 18 terminates after one iteration, or the set C_f is feasible and it holds that* $\mathrm{NCut}(C, \overline{C}) > \mathrm{NCut}(C_f, \overline{C_f})$.

Proof. This is a direct corollary of Theorem 5.13. □

8.3.1 Solution of the inner problem

The crucial step in the algorithm is solving the inner problem (line 5). We solve this problem via an equivalent smooth dual problem. The first step is to eliminate the norm constraint in the objective.

Lemma 8.14. *Let Φ be a 1-homogeneous function and $\|\cdot\|$ any norm, and*

$$f^* \in \operatorname*{arg\,min}_{x \geq 0} \Phi(x) + \rho \|x\|^2,$$

for any $\rho > 1$. If $f^* \neq 0$, then set $f' := \frac{f^*}{\|f^*\|}$, otherwise set $f' := 0$. Then,

$$f' \in \operatorname*{arg\,min}_{\|x\| \leq 1, x \geq 0} \Phi(x). \tag{8.14}$$

Proof. We first assume $f^* \neq 0$. Then one has

$$\begin{aligned}
\Phi\left(\tfrac{f^*}{\|f^*\|}\right) &= \tfrac{1}{\|f^*\|}\left(\Phi\left(f^*\right) + \rho \|f^*\|^2\right) - \rho \|f^*\| \\
&= \tfrac{1}{\|f^*\|}\left(\min_{x \geq 0} \Phi(x) + \rho \|x\|^2\right) - \rho \|f^*\| \\
&\leq \tfrac{1}{\|f^*\|}\left(\min_{x \geq 0,\, \|x\| = \|f^*\|} \Phi(x) + \rho \|x\|^2\right) - \rho \|f^*\| \\
&= \min_{x \geq 0,\, \|x\| = \|f^*\|} \Phi\left(\tfrac{x}{\|x\|}\right) + \rho \|f^*\| - \rho \|f^*\| \\
&= \min_{y \geq 0,\, \|y\| = 1} \Phi(y),
\end{aligned}$$

which with the 1-homogeneity of Φ implies that

$$\tfrac{f^*}{\|f^*\|} \in \operatorname*{arg\,min}_{\|y\|=1,\, y\geq 0} \Phi(y) = \operatorname*{arg\,min}_{\|y\|\leq 1,\, y\geq 0} \Phi(y).$$

Now let $f^* = 0$ and assume for the sake of contradiction that $\Phi(f') < 0$. Due to the one-homogeneity this implies that $\|f'\| = 1$. Let $x' = \alpha f'$, for some $\alpha \in (0, -\frac{1}{\rho}\Phi(f'))$. Then, using the homogeneity of Φ and $\|\|$, one has

$$\Phi(x') + \rho \left\|x'\right\|^2 = \alpha\Phi(f') + \alpha^2\rho \left\|f'\right\|^2 = \alpha\left(\Phi(f') + \alpha\rho\right) < 0,$$

which is a contradiction to $f^* = 0$ being optimal. Thus $\Phi(f') = 0$ and hence $f' = 0$ is a minimizer of the problem in (8.14). □

Lemma 8.14 implies that the inner problem (8.11) can be replaced by

$$\min_{f\in\mathbb{R}^m_+} \tfrac{1}{2}\sum_{i,j=1}^{m} w_{ij}\left|f_i - f_j\right| + c_1 \max_i f_i + \left\langle f, c_2^k\right\rangle + \tfrac{1}{2}\|f\|_2^2 . \tag{8.15}$$

Then, given a solution f^* of (8.15), a solution f' of (8.11) is obtained via $f' = f^*/\|f^*\|_2$, if $f^* \neq 0$, otherwise we set $f' = 0$.

In the following, for $m \in \mathbb{N}$, $P_{\mathbb{R}^m_+}$ denotes the projection on the positive orthant and S_m is the simplex $S_m = \{v \in \mathbb{R}^m \,|\, v_i \geq 0, \sum_{i=1}^m v_i = 1\}$. One can now derive the dual problem as follows.

Lemma 8.15. *The inner problem* (8.15) *is equivalent to*

$$-\min_{\substack{\alpha\in\mathbb{R}^{|E|}\\ \|\alpha\|_\infty\leq 1}} \min_{v\in S_m} \Psi(\alpha) := \left\|P_{\mathbb{R}^m_+}\big(-c_1 v - c_2^k - A\alpha\big)\right\|_2^2$$

where the operator $A : \mathbb{R}^{|E|} \to \mathbb{R}$ *is for* $\alpha \in \mathbb{R}^{|E|}$ *defined as* $(A\alpha)_i := \frac{1}{2}\sum_{j\,|\,(i,j)\in E} w_{ij}(\alpha_{ij} - \alpha_{ji})$. *The gradient of* Ψ *is given as*

$$(\nabla\Psi(\alpha))_{rs} = -w_{rs}\,(z_r - z_s)\,, \qquad \text{where } z = P_{\mathbb{R}^m_+}\big(-c_1 v - c_2^k - A\alpha\big).$$

Moreover, an upper bound on the Lipschitz constant of the gradient of Ψ *is given by* $2\max_r \sum_{s\,|\,(r,s)\in E} w_{rs}^2$.

Proof. We derive the dual problem as follows:

$$\begin{aligned}
&\min_{f\in\mathbb{R}^m_+} \tfrac{1}{2}\sum_{i,j=1}^{m} w_{ij}\left|f_i - f_j\right| + c_1 \max_i f_i + \left\langle f, c_2^k\right\rangle + \tfrac{1}{2}\|f\|_2^2 \\
&= \min_{f\in\mathbb{R}^m_+} \Big\{ \max_{\substack{\alpha\in\mathbb{R}^{|E|}\\ \|\alpha\|_\infty\leq 1}} \tfrac{1}{2}\sum_{(i,j)\in E} w_{ij}\,(f_i - f_j)\,\alpha_{ij} \\
&\qquad\qquad + \max_{v\in S_m} c_1\langle f, v\rangle \;+\; \left\langle f, c_2^k\right\rangle \;+\; \tfrac{1}{2}\|f\|_2^2 \Big\} \\
&= \max_{\substack{\alpha\in\mathbb{R}^{|E|}\\ \|\alpha\|_\infty\leq 1}} \max_{v\in S_m} \min_{f\in\mathbb{R}^m_+} \tfrac{1}{2}\|f\|_2^2 \;+\; \left\langle f, c_1 v + c_2^k + A\alpha\right\rangle,
\end{aligned}$$

where $(A\alpha)_i := \frac{1}{2}\sum_{j\,|\,(i,j)\in E} w_{ij}(\alpha_{ij} - \alpha_{ji})$. The optimization over f has the solution $f = P_{\mathbb{R}^m_+}\left(-c_1 v - c_2^k - A\alpha\right)$. Plugging f into the objective and using that $\left\langle P_{\mathbb{R}^m_+}(x), x\right\rangle = \left\|P_{\mathbb{R}^m_+}(x)\right\|_2^2$, we obtain the first statement. Regarding the gradient, note that for $x \in \mathbb{R}^n$, $\nabla\frac{1}{2}\left\|P_{\mathbb{R}^m_+}(x)\right\|_2^2 = P_{\mathbb{R}^m_+}(x)$, as shown in Lemma 6.6. Thus we obtain

$$\begin{aligned}(\nabla\Psi(\alpha))_{rs} &= \sum_{i\in V} 2\Big(P_{\mathbb{R}^m_+}\big(-c_1 v - c_2^k - A\alpha\big)\Big)\cdot\Big(-\tfrac{1}{2}w_{is}\delta_{i=r} + \tfrac{1}{2}w_{ir}\delta_{i=s}\Big)\\ &= -w_{rs}\Big(P_{\mathbb{R}^m_+}\big(-c_1 v - c_2^k - A\alpha\big)_r - P_{\mathbb{R}^m_+}\big(-c_1 v - c_2^k - A\alpha\big)_s\Big).\end{aligned}$$

We now derive an upper bound on the Lipschitz constant of the gradient. Introducing the notation $z(\alpha) := -c_1 v - c_2^k - A\alpha$, we obtain for $\alpha, \alpha' \in \mathbb{R}^{|E|}$,

$$\begin{aligned}&\left\|\nabla\Psi(\alpha) - \nabla\Psi(\alpha')\right\|_2^2\\ &= \sum_{(r,s)\in E} w_{rs}^2\Big(P_{\mathbb{R}^m_+}\big(z(\alpha)\big)_r - P_{\mathbb{R}^m_+}\big(z(\alpha')\big)_r - P_{\mathbb{R}^m_+}\big(z(\alpha)\big)_s + P_{\mathbb{R}^m_+}\big(z(\alpha')\big)_s\Big)^2\\ &\le 4\sum_{(r,s)\in E} w_{rs}^2\Big(P_{\mathbb{R}^m_+}\big(z(\alpha)\big)_r - P_{\mathbb{R}^m_+}\big(z(\alpha')\big)_r\Big)^2,\end{aligned}$$

where we used the fact that for $a, b \in \mathbb{R}$ it holds that $(a-b)^2 \le 2a^2 + 2b^2$, as well as the symmetry of W. Using the fact that it holds for $x, y \in \mathbb{R}$ that $(\max\{x,0\} - \max\{y,0\})^2 \le (x-y)^2$, we obtain

$$\ldots \;\le 4\sum_{(r,s)\in E} w_{rs}^2\Big(z(\alpha)_r - z(\alpha')_r\Big)^2 = 4\sum_{(r,s)\in E} w_{rs}^2\Big((A\alpha)_r - (A\alpha')_r\Big)^2.$$

One can now proceed as in the proof of Lemma 7.38 to show that

$$\left\|\nabla\Psi(\alpha) - \nabla\Psi(\alpha')\right\|_2^2 \;\le\; 4\Big(\max_r \sum_{s\,|\,(r,s)\in E} w_{rs}^2\Big)^2\;\left\|\alpha - \alpha'\right\|_2^2,$$

which implies that an upper bound on the Lipschitz constant is given by $2\max_r \sum_{s\,|\,(r,s)\in E} w_{rs}^2$. □

This dual problem can be solved efficiently using Nesterov's method [Nesterov, 1983, Beck and Teboulle, 2009], see Section 6. The explicit steps are given in Alg. 19 (we again use the notation $B_\infty(1) := \{x \in \mathbb{R}\,|\,|x| \le 1\}$).

Note that as in the case of 1-spectral clustering, the memory requirement can be further reduced by only considering the upper triangular part of α and enforcing anti-symmetry in each step, see the discussion after Alg. 16. To solve the first subproblem, one can make use of the following fact.

Lemma 8.16. *Let $x \in \mathbb{R}^m$ and $y := P_{\mathbb{R}^m_+}(x)$, then*

$$\operatorname*{arg\,min}_{v\in S_m}\left\|y - v\right\|_2^2 \in \operatorname*{arg\,min}_{v\in S_m}\left\|P_{\mathbb{R}^m_+}\left(x - v\right)\right\|_2^2.$$

19 Solution of the dual inner problem with Nesterov's method

Input: Lipschitz constant L of $\nabla\Psi$,
Initialization: $\theta^0 = 1$, $\alpha^0 \in \mathbb{R}^{|E|}$,
repeat
$v^t = \arg\min_{u \in S_m} \left\| P_{\mathbb{R}^m_+}\left(- c_1 u - c_2^k - A\alpha^t\right)\right\|_2^2$
$z^t = P_{\mathbb{R}^m_+}\left(-c_1 v^t - c_2^k - A\alpha^t\right)$
$\beta_{rs}^{t+1} = P_{B_\infty(1)}\left(\alpha_{rs}^t + \frac{1}{L} w_{rs}\left(z_r^t - z_s^t\right)\right)$
$\theta^{t+1} = \frac{1+\sqrt{1+4(\theta^t)^2}}{2}$,
$\alpha_{rs}^{t+1} = \beta_{rs}^{t+1} + \frac{\theta^t - 1}{\theta^{t+1}}\Big(\beta_{rs}^{t+1} - \beta_{rs}^t\Big)$.
until duality gap $< \epsilon$

Proof. First observe that $\forall v \in S_m$,

$$\begin{aligned}
&\left\| P_{\mathbb{R}^m_+}(y - v)\right\|_2^2 \;=\; \sum_{i=1}^m \max\left\{\max\{x_i, 0\} - v_i, 0\right\}^2 \\
&= \sum_{\max\{x_i,0\} > v_i} \left(\max\{x_i, 0\} - v_i\right)^2 \;=\; \sum_{x_i > v_i} \left(\max\{x_i, 0\} - v_i\right)^2 \\
&= \sum_{x_i > v_i} \left(x_i - v_i\right)^2 \;=\; \left\| P_{\mathbb{R}^m_+}(x - v)\right\|_2^2,
\end{aligned}$$

where in the third and fourth step we have used the fact that $v_i \geq 0, \forall i = 1 \ldots m$. Hence one has to show that

$$\arg\min_{v \in S_m} \tfrac{1}{2}\left\|y - v\right\|_2^2 \in \arg\min_{v \in S_m} \tfrac{1}{2}\left\| P_{\mathbb{R}^m_+}(y - v)\right\|_2^2.$$

The Lagrangian of the left side is given as

$$L(v, \gamma, \mu) = \tfrac{1}{2}\left\|y - v\right\|_2^2 - \sum_{i=1}^m \gamma_i v_i + \mu\Big(\sum_{i=1}^m v_i - 1\Big).$$

We now derive the KKT conditions. One obtains the stationarity condition

$$v_i - y_i - \gamma_i + \mu = 0, \qquad \forall i = 1 \ldots m, \tag{8.16}$$

as well as $-v_i \leq 0$, $\forall i = 1 \ldots m$ and $\sum_{i=1}^m v_i = 1$ for primal feasibility, and $\gamma_i \geq 0$, $\forall i = 1 \ldots m$ for dual feasibility. Moreover, the complementary slackness condition is given as $\gamma_i v_i = 0$, $\forall i = 1 \ldots m$. On the other hand, the Lagrangian of the right side is given as

$$L(v, \gamma, \mu) = \tfrac{1}{2}\left\| P_{\mathbb{R}^m_+}(y - v)\right\|_2^2 - \sum_{i=1}^m \gamma_i v_i + \mu\Big(\sum_{i=1}^m v_i - 1\Big).$$

Here one obtains the same KKT conditions as for the left side, except for the first condition in (8.16), which becomes

$$-\left(\max\left\{y_i - v_i, 0\right\}\right) - \gamma_i + \mu = 0, \qquad \forall i = 1 \dots m.$$

This can be rewritten as

$$\begin{aligned} v_i - y_i - \gamma_i + \mu &= 0, && \forall i = 1 \dots m,\ y_i \geq v_i, \\ -\gamma_i + \mu &= 0, && \forall i = 1 \dots m,\ y_i < v_i. \end{aligned}$$

Let (v, γ, μ) satisfy the KKT conditions of the left side. We first treat the case when $\mu \geq 0$. Note that $v_i \geq 0, \forall i = 1 \dots m$. If $v_i = 0$, we have $y_i = \max\{x_i, 0\} \geq v_i$. On the other hand, if $v_i > 0$, the complementary slackness condition $v_i \gamma_i = 0$ implies $\gamma_i = 0$, and therefore the KKT condition (8.16) implies that $y_i - v_i = \mu \geq 0$. Hence in both cases we have $y_i \geq v_i$, which implies that (v, γ, μ) satisfies the KKT conditions of the right side.

For the case $\mu < 0$, assume that $\exists k \in \{1 \dots m\}$ with $v_k = 0$. Then on the one hand it holds due to (8.16) that $y_k + \gamma_k = \mu < 0$, on the other hand one has $y_k + \gamma_k \geq 0$, as $\gamma_i \geq 0$ and $y_i \geq 0,\ \forall i = 1 \dots m$, which is a contradiction. Thus $v_i > 0, \forall i = 1 \dots m$. The complementary slackness condition again implies that $\gamma_i = 0$, and therefore the KKT condition (8.16) yields $y_i < v_i$. One now checks easily that $(v, \gamma, 0)$ satisfies the KKT conditions of the right side. □

Lemma 8.16 implies that the minimization problem in the first line of Alg. 19 can be solved via a standard projection onto the simplex, which can be computed in linear time [Kiwiel, 2007]. Thus the most expensive part of each iteration of Alg. 19 is the sparse matrix multiplication $A\alpha$, which scales linearly in the number of edges.

8.4 Experimental results

In our first experiment, we evaluate our tight relaxation of the constrained NCut with volume and seed constraint on a number of large social networks. We use the version of our method where the seed constraint is directly incorporated into the objective, see Section 8.2.2. In the second experiment, we consider the variant of our algorithm where both seed constraint and volume constraint are treated as penalties according to Section 8.2.3, and investigate its usefulness to deal with soft or noisy subset and volume constraints.

8.4.1 Social networks

We first evaluate our approach on several large networks of the Stanford Large Network Dataset Collection [Leskovec]. We use two collaboration networks (CA-GrQc and CA-HepTh), two citation networks (Cit-HepTh and

Cit-HepPh) as well as two product co-purchasing networks (amazon0302 and amazon0505). The directed graphs were transformed into undirected graphs by adding a back-edge for every outgoing edge. Moreover, we extracted the largest connected component for each graph.

As optimization criterion we use the local normalized cut with a subset constraint and a volume constraint of the form $\mathrm{vol}(C) \leq k$. We compare our method (denoted as CFSP) against the Local Spectral (LS) method by Mahoney et al. [2012] and the Lazy Random Walk (LRW) by Andersen and Lang [2006]. The code of Hansen and Mahoney [2012] is used to compute the solution of LS in our experiments. In the case of the LRW, for a fair comparison, we compute the full sequence of random walk vectors until the stationary distribution is reached, and in each step perform constrained optimal thresholding according to the normalized cut objective. In the case of the RatioDCA we perform 10 runs with different random initializations and report the result with smallest objective value. Regarding the parameter γ from Theorem 4.6, it turns out that best results are obtained by first solving the unconstrained case ($\gamma = 0$) and then increasing γ sequentially, until all constraints are fulfilled.

For each dataset we generate 10 random seeds. In order to ensure that meaningful intervals for the volume constraint are explored, we first solve the local clustering problem only with the seed constraint. Treating this as the "unconstrained" solution C_0, we then repeat the experiment with upper bounds of the form $\mathrm{vol}(C) \leq \alpha \,\mathrm{vol}(C_0)$, where $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$. In this way we ensure that for each dataset, meaningful intervals for the volume constraint are explored.

Table 8.1 shows mean and standard deviation of the NCut values averaged over the 10 different random trials (seeds) and average runtime over the different seeds and volume constraints. Our method CFSP consistently outperforms the competing methods by large margins and always finds solutions that satisfy all constraints. The method LRW is very fast, but the obtained normalized cuts are not competitive. Note that CFSP still performs better if one thresholds the obtained solutions according to the normalized Cheeger cut for which LRW has been designed. This is shown in Table 8.2 where we compare the normalized Cheeger cut of our solutions (note that we optimized the normalized cut) to the solution obtained by the Lazy Random Walk method where we threshold in each step according to the normalized Cheeger cut objective.

Table 8.1: Results for the constrained local normalized cut on large social networks. Our solutions (CFSP) always satisfy all constraints and have smaller cuts than the two competing methods LS and LRW.

Dataset	Method	$\leq 20\%$	$\leq 40\%$	$\leq 60\%$	$\leq 80\%$	$\leq 100\%$	Avg. time
CA-GrQc	LRW	0.1311 (0.0686)	0.1005 (0.0542)	0.0984 (0.0543)	0.0920 (0.0439)	0.0773 (0.0341)	2
$\|V\| = 4158$	LS	0.2014 (0.0958)	0.1182 (0.0958)	0.0685 (0.1089)	0.0314 (0.0423)	0.0217 (0.0259)	6
$\|E\| = 13422$	CFSP	**0.0315 (0.0292)**	**0.0157 (0.0131)**	**0.0138 (0.0115)**	**0.0083 (0.0055)**	**0.0069 (0.0044)**	31
CA-HepTh	LRW	0.2607 (0.0914)	0.2157 (0.0533)	0.2015 (0.0498)	0.1954 (0.0491)	0.1888 (0.0483)	9
$\|V\| = 8638$	LS	0.4125 (0.1079)	0.3439 (0.0631)	0.3089 (0.0839)	0.2926 (0.0913)	0.2778 (0.0923)	13
$\|E\| = 24806$	CFSP	**0.0518 (0.0226)**	**0.0327 (0.0104)**	**0.0318 (0.0094)**	**0.0263 (0.0082)**	**0.0104 (0.0038)**	58
Cit-HepTh	LRW	0.5052 (0.2208)	0.4697 (0.2010)	0.4373 (0.1962)	0.4067 (0.1998)	0.3807 (0.2224)	15
$\|V\| = 27400$	LS	0.5430 (0.2617)	0.5099 (0.2524)	0.4737 (0.2586)	0.4290 (0.2773)	0.3997 (0.2834)	175
$\|E\| = 352021$	CFSP	**0.4693 (0.2676)**	**0.3732 (0.2166)**	**0.2683 (0.1494)**	**0.1748 (0.0683)**	**0.0752 (0.0233)**	3704
Cit-HepPh	LRW	0.1784 (0.0541)	0.1466 (0.0503)	0.1234 (0.0256)	0.1079 (0.0120)	0.1048 (0.0062)	19
$\|V\| = 34401$	LS	0.1720 (0.0055)	0.1292 (0.0224)	0.1155 (0.0147)	0.1107 (0.0062)	0.1078 (0.0007)	103
$\|E\| = 420784$	CFSP	**0.1181 (0.0143)**	**0.1127 (0.0101)**	**0.1109 (0.0089)**	**0.0928 (0.0039)**	**0.0913 (0.0015)**	2666
amazon0302	LRW	0.1768 (0.0833)	0.1465 (0.0749)	0.1336 (0.0601)	0.1221 (0.0504)	0.1120 (0.0429)	336
$\|V\| = 262111$	LS	0.2662 (0.1204)	0.2496 (0.1155)	0.2247 (0.1021)	0.2066 (0.0892)	0.1946 (0.0840)	5765
$\|E\| = 899792$	CFSP	**0.0194 (0.0063)**	**0.0095 (0.0043)**	**0.0072 (0.0031)**	**0.0056 (0.0024)**	**0.0050 (0.0022)**	3007
amazon0505	LRW	0.2472 (0.1112)	0.2369 (0.1124)	0.2249 (0.1132)	0.2200 (0.1152)	0.2163 (0.1183)	210
$\|V\| = 410236$	LS	0.4124 (0.1751)	0.3704 (0.1864)	0.3653 (0.1878)	0.3576 (0.1919)	0.3529 (0.1956)	20558
$\|E\| = 2439437$	CFSP	**0.0227 (0.0076)**	**0.0116 (0.0089)**	**0.0058 (0.0020)**	**0.0048 (0.0011)**	**0.0047 (0.0008)**	13171

Table 8.2: Constrained local normalized Cheeger cuts of the solutions obtained by our method on large social networks (note that we optimized the normalized cut) as well as the solutions of Lazy Random Walk (LRW) where we threshold in each step according to the normalized Cheeger cut objective.

Dataset	Method	$\leq 20\%$	$\leq 40\%$	$\leq 60\%$	$\leq 80\%$	$\leq 100\%$	Avg. time
CA-GrQc	LRW	0.1298 (0.0677)	0.0992 (0.0536)	0.0967 (0.0537)	0.0894 (0.0418)	0.0753 (0.0340)	1
	CFSP	**0.0312 (0.0289)**	**0.0153 (0.0128)**	**0.0133 (0.0110)**	**0.0079 (0.0051)**	**0.0064 (0.0040)**	31
CA-HepTh	LRW	0.2601 (0.0911)	0.2150 (0.0530)	0.2005 (0.0495)	0.1941 (0.0488)	0.1873 (0.0481)	1
	CFSP	**0.0517 (0.0225)**	**0.0326 (0.0104)**	**0.0317 (0.0093)**	**0.0261 (0.0082)**	**0.0103 (0.0037)**	58
Cit-HepTh	LRW	0.4967 (0.2300)	0.4565 (0.2150)	0.4179 (0.2174)	0.3890 (0.2174)	0.3705 (0.2307)	10
	CFSP	**0.4673 (0.2690)**	**0.3712 (0.2176)**	**0.2661 (0.1496)**	**0.1681 (0.0706)**	**0.0705 (0.0150)**	3704
Cit-HepPh	LRW	0.1574 (0.0497)	0.1104 (0.0364)	**0.0769 (0.0151)**	0.0573 (0.0064)	**0.0566 (0.0062)**	14
	CFSP	**0.1168 (0.0156)**	**0.1067 (0.0138)**	0.0986 (0.0202)	**0.0500 (0.0098)**	0.0584 (0.0049)	2666
amazon0302	LRW	0.1768 (0.0833)	0.1464 (0.0749)	0.1335 (0.0600)	0.1220 (0.0503)	0.1118 (0.0428)	241
	CFSP	**0.0193 (0.0063)**	**0.0095 (0.0043)**	**0.0072 (0.0031)**	**0.0056 (0.0024)**	**0.0050 (0.0022)**	3007
amazon0505	LRW	0.2472 (0.1111)	0.2369 (0.1124)	0.2248 (0.1132)	0.2200 (0.1152)	0.2162 (0.1183)	289
	CFSP	**0.0227 (0.0076)**	**0.0116 (0.0089)**	**0.0058 (0.0020)**	**0.0048 (0.0011)**	**0.0047 (0.0008)**	13171

8.4.2 Weak or noisy constraints

By Theorem 4.7, the solution of the tight relaxation is guaranteed to satisfy all constraints. This is a desirable property assuming that the constraints represent available hard knowledge about the problem. However, there are also situations where this is not the case. For instance, the seed constraints could be generated by a human or automatically derived from some measurements, and thus in both cases be susceptible to errors or random fluctuations. Moreover, the upper bound on the volume could be only an approximate estimate instead of a hard constraint. Thus in both cases one is willing to sacrifice some of the constraints if it leads to a better NCut objective. In the case of the noisy constraints, it is even desired to omit some of the constraints. For this reason, in this experiment we consider the variant of our algorithm where the subset and volume constraint are handled via penalty terms and show that by adjusting the penalty parameters γ_1 and γ_2 one obtains a useful technique to deal with noisy and weak constraints.

We construct an instance of the noisy two moons dataset (see Section 7) as two half moons of n points in 2 dimensions embedded in a d-dimensional space, where in this case $d = 20$ and $n = 1000$. In contrast to Section 7, the number of points in the two moons is heavily imbalanced (777 points in upper and 223 points in lower moon). Moreover, also the noise variance differs, it is given as 0.04 for the upper moon and 0.02 for the lower moon.

Due to the fact that there are many edges between the parts of the graph corresponding to the right part of the upper moon and the left part of the lower moon, the partition obtaining the best NCut (without any constraints) cuts through the upper cluster, see Fig. 8.3.

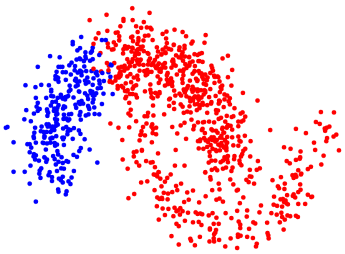

Figure 8.3: Optimal (unconstrained) NCut on unbalanced two moons dataset (Upper moon: 777 points, noise variance 0.04. Lower moon: 223 points, noise variance 0.02).

Thus in this case optimizing NCut without any constraints is not suitable to separate the two half moons. We now consider the scenario where we are given some additional information in the form of constraints which allow us

to obtain a better clustering. We first consider the case where the constraints are noise-free and specify some seed vertices (around 4% of the total number of vertices) in the lower half moon, see the top left plot in Fig. 8.4. We run our algorithm for the local normalized cut with the given seed set and a volume constraint $\mathrm{vol}(C) \leq 0.25\,\mathrm{vol}(V)$. The resulting clustering is shown in the top right plot in Fig. 8.4. As one can see, now the two half moons are separated almost perfectly.

Next we consider the case where the constraints are noisy. Thus we take the seed set from the previous run and randomly add some vertices to the seed set (with 5% probability). The resulting seed set, which is shown in the bottom left of Fig. 8.4, now contains 35 vertices in the upper moon and 53 vertices in the lower moon. Moreover, we consider a volume constraint $\mathrm{vol}(C) \leq 0.15\,\mathrm{vol}(V)$. Note that this constraint is too strict since the volume of the lower moon is about one fourth of the total volume. As one can see in the bottom right of Fig. 8.4, enforcing the constraints does not lead to a meaningful result in this case.

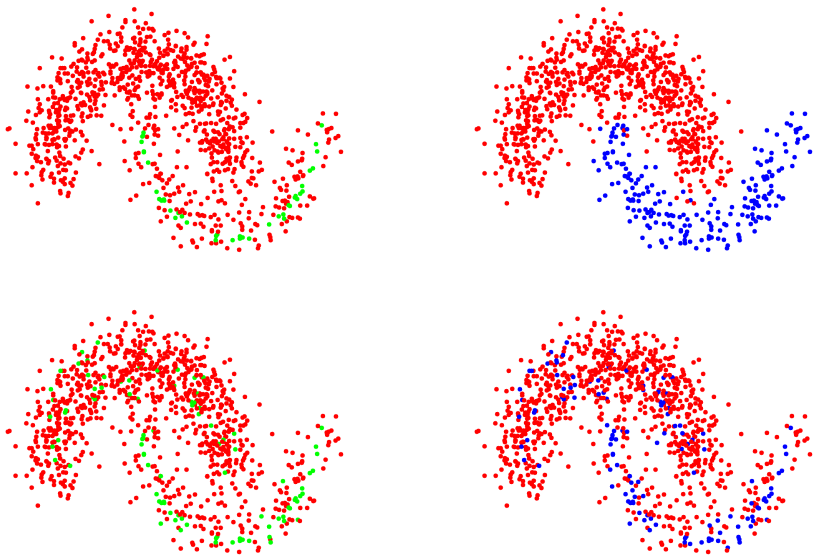

Figure 8.4: Constrained local clustering (with hard subset and volume constraint) on the unbalanced two moons data set. *Top row:* Results for noise-free seed set and volume constraint $\mathrm{vol}(C) \leq 0.25\,\mathrm{vol}(V)$. *Left*: Seed set. *Right*: Obtained partition. *Bottom row:* Results for noisy seed set and (too strict) volume constraint $\mathrm{vol}(C) \leq 0.15\,\mathrm{vol}(V)$. *Left:* Seed set. *Right:* Suboptimal partition obtained when strictly enforcing constraints.

We now apply our penalty-based algorithm to the same problem. One obtains two penalty terms $\widehat{T}_1(C)$ and $\widehat{T}_2(C)$ with parameters γ_1 and γ_2 for

the subset constraint and the volume constraint, respectively. Since there is a big difference in the range of values of the penalty terms $\widehat{T}_1(C)$ and $\widehat{T}_2(C)$, we add an additional scaling term to the penalty term for the seed constraint, which then appears in the objective as $\gamma_1\, \alpha\, \widehat{T}_1(C)$, where

$$\alpha = \frac{\max_A \widehat{T}_2(A)}{\max_A \widehat{T}_1(A)} = \frac{\frac{1}{2}\operatorname{vol}(V) - k_2}{|J|}.$$

We then run our algorithm for different values of γ_1, γ_2 between 0 (unconstrained solution) and 1.

In the left plot of Figure 8.5 we give the relative number of seed vertices contained in the resulting cluster for different values of $\gamma_1, \gamma_2 \in [0, 1]$. For $\gamma_1 = 0$ less than 20% of the seed vertices are contained in the solution. This number becomes larger for increasing values of γ_1. For $\gamma_1 \geq 0.4$, all seed vertices are contained in the solution. The middle plot shows the volumes of the set C relative to the upper bound. Note that here the axis corresponding to γ_2 has been reversed. One observes that while for $\gamma_2 = 0$, the volume constraint is violated by almost a factor 3, increasing the value of γ_2 leads to a decrease in volume. For $\gamma_2 \geq 0.8$, the volume constraint is satisfied for all values of γ_1. Moreover, one also observes a small increase in the γ_1 direction, as for higher values of γ_1 (more seeds contained in the cluster), it becomes harder to enforce the volume constraint. For $\gamma_1 = 0$, the volume constraint is fulfilled for all values $\gamma_2 > 0$. However, enforcing the constraints comes at the cost of higher objective values. This is shown in the right plot where we display the achieved NCut values. One observes a significant increase in NCut for increasing values of γ_1 and γ_2. Clearly, the smallest NCut value is obtained for $\gamma_1 = \gamma_2 = 0$ (unconstrained solution).

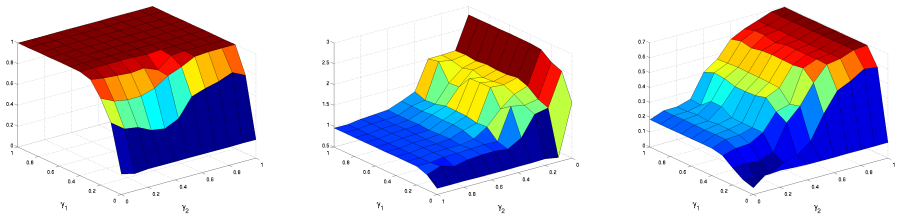

Figure 8.5: The effect of choosing different values of γ_1 and γ_2. *Left:* Relative number of seed vertices in the resulting cluster. *Middle:* Volumes of set C relative to upper bound. *Right:* NCut values.

In Figure 8.6 we show the clustering obtained for the noisy seed set (shown in the bottom left) in the four extreme cases: For $\gamma_1 = \gamma_2 = 0$ (top left), one obtains the unconstrained solution. For $\gamma_1 = 0$ and $\gamma_2 = 1$ (top right), the seed constraint is ignored but the volume constraint is enforced,

thus the resulting cluster is a small part of the lower moon. The two plots in the second row of Figure 8.6 show the case when the seed constraint is enforced ($\gamma_1 = 1$). In the left plot, the volume constraint is ignored ($\gamma_2 = 0$), thus the resulting cluster consists of the lower right part of the dataset plus the additional seed vertices. In the right plot, the volume constraint is enforced. Since the volume of the seed set is already about 10% of the total volume, the resulting cluster is not meaningful. Thus, in order to separate the two half moons, one needs to choose values of γ_1 and γ_2 between these extreme cases. The best results are obtained with $\gamma_1 = 0.1$ and $\gamma_2 = 0.2$, see the bottom right plot in Figure 8.6.

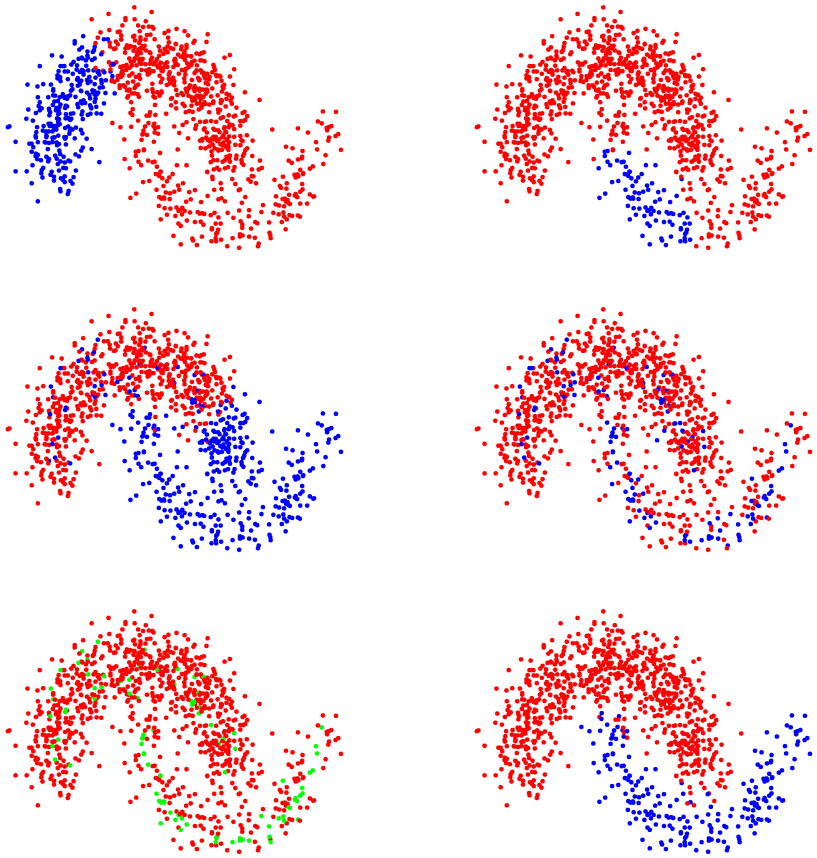

Figure 8.6: Constrained local clustering (with subset and volume constraints via penalties) for noisy seeds and volume constraint $\mathrm{vol}(C) \leq 0.15\,\mathrm{vol}(V)$. *First row, left:* $\gamma_1 = 0$, $\gamma_2 = 0$. *Right:* $\gamma_1 = 0$, $\gamma_2 = 1$. *Second row, left:* $\gamma_1 = 1$, $\gamma_2 = 0$. *Right:* $\gamma_1 = 1$, $\gamma_2 = 1$. *Third row, left*: Noisy seed set. *Right*: Best result is obtained for $\gamma_1 = 0.1$ and $\gamma_2 = 0.2$.

Chapter 9

Community detection via the densest subgraph problem

The maximum density subgraph problem is an important problem in computer science, which has a wide range of applications in graph analysis, for instance in finding substructures in web graphs or social networks [Dourisboure et al., 2007], spam detection [Gibson et al., 2005] or bioinformatics [Saha et al., 2010]. In this chapter we will use the results from Chapter 4 to derive a reformulation of the (generalized) maximum density subgraph problem, which can then be solved using the RatioDCA introduced in Chapter 5. We will demonstrate the usefulness of the approach for community detection on two social networks.

9.1 The constrained densest subgraph problem

In contrast to the local clustering problem discussed in the last chapter, in community detection it makes more sense to find a highly connected set instead of minimizing the separation to the remaining part of the graph. In the following, we are again in the same graph-based setting as for the previous applications, see Section 7.1. The classical densest subgraph problem (with seed subset) can then be formulated as

$$\begin{aligned} &\max_{C \subset V} \frac{\operatorname{assoc}(C)}{|C|} \\ &\text{subject to} : J \subset C, \end{aligned}$$

where $\operatorname{assoc}(C) = \sum_{i,j \in C} w_{ij}$. In Figure 9.1 we show an example of a dense subgraph on a graph. Here we have $\operatorname{assoc}(C) = 18$ (note that each edge is counted twice) and $|C| = 3$ and thus the density of the set C is 6. One can easily see that adding any vertex to the set would decrease the density, thus the subgraph is optimal for the given graph. For comparison, we have $\operatorname{assoc}(V) = 40$ and $|V| = 8$ and thus the density of the whole graph is 5.

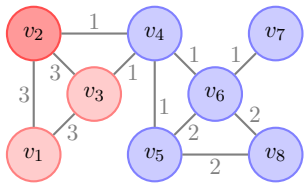

Figure 9.1: Densest subgraph (red) with subset constraint

The classical (unconstrained) densest subgraph problem can be solved optimally in polynomial time [Goldberg, 1984]. Moreover, one can compute a 2-approximation in linear time [Charikar, 2000]. However, a significantly more difficult problem is the *densest-k-subgraph problem*, given as

$$\max_{C \subset V} \frac{\text{assoc}(C)}{|C|}$$
$$\text{subject to} : |C| = k, \quad \text{and} \quad J \subset C.$$

Here one requires the solution to contain exactly k vertices. Adding the size constraint makes the problem NP hard [Feige et al., 2001], and it has been shown not to admit a polynomial time approximation scheme [Khot, 2006], i.e. there is no polynomial time algorithm which produces a solution within a factor of $1 - \varepsilon$ of the optimal solution for arbitrary small constant $\varepsilon > 0$. Moreover, the best known approximation algorithm for general k has an approximation ratio of $O(|V|^{\delta})$, for some $\delta < \frac{1}{3}$ [Feige et al., 2001].

Replacing the equality constraint with a *lower bound constraint* (shown below) leads to a problem which is still NP hard [Khuller and Saha, 2009], but has a 2-approximation algorithm [Andersen, 2007, Khuller and Saha, 2009]. Problems of this form have been considered in team selection [Gajewar and Das Sarma, 2012] and bioinformatics [Saha et al., 2010].

$$\max_{C \subset V} \frac{\text{assoc}(C)}{|C|}$$
$$\text{subject to} : k \leq |C|, \quad \text{and} \quad J \subset C.$$

However, if one has an *upper bound constraint* on the size, the problem is as hard as the densest-k-subgraph problem [Andersen and Chellapilla, 2009, Khuller and Saha, 2009]. In particular, it has been shown by Khuller and Saha [2009] that the existence of an approximation algorithm with approximation ratio α for the *densest-at most k-subgraph* problem implies that there is an approximation algorithm for the densest-k-subgraph problem with approximation ratio 4α. In the following, we will consider a generalized version

of the densest subgraph problem,

$$\begin{aligned} \max_{C \subset V} \frac{\text{assoc}(C)}{\text{vol}_g(C)} &:= \text{Density}_g(C) \\ \text{subject to} : k_1 &\leq \text{vol}_{h_1}(C), \\ \text{vol}_{h_2}(C) &\leq k_2, \quad \text{and} \quad J \subset C, \end{aligned} \tag{9.1}$$

where $g, h_1, h_2 \in \mathbb{R}^n_+$ are general non-negative vertex weights. This formulation incorporates all previously discussed problems as special cases. Moreover, it is a further generalization as it replaces the cardinalities in the constraints as well as the denominator by general volume functions vol_g and vol_{h_1}, vol_{h_2}. Given a seed set $J \subset C$ consisting of one or more seed vertices, the above formulation allows us to find dense communities containing the seed set and thus can be used to analyze the given community structure in a graph, for instance a social network. Furthermore, the use of vertex weights in the denominator allows us to bias the obtained community towards one with desired properties (assigning small weights to vertices which one prefers to be contained in the solution, larger weights to ones which are less preferred). In the experiments we will consider an example where we detect communities in a network of computer science authors.

Another application is to form teams in a social network, as shown in [Rangapuram et al., 2013]. Here one considers a graph which models the mutual compatibility between a set of experts (for instance constructed based on previous collaboration between the experts) and the goal is to identify a team of experts (a subset of the graph) which is highly collaborative as well as competent to perform a given task. The (generalized) density then yields a useful criterion for the collaborative compatibility of the team. While there exist other possibilities to measure the collaborative compatibility, for instance using the diameter of the subgraph, the cost of the minimum spanning tree Lappas et al. [2009], or the shortest path distances between the experts Kargar and An [2011], the density has several useful properties, for example strict monotonicity and robustness [Gajewar and Das Sarma, 2012].

The constraints allow us to incorporate additional requirements on the obtained community into the optimization problem. For instance, in the team formation application, each team member, or the team as a whole, may need to satisfy a certain skill requirement, and usually there are budget constraints or a bound on the team size. Moreover, typically the team leader is fixed in advance, which quite naturally leads to a subset constraint. It was shown in [Rangapuram et al., 2013] that the additional flexibility obtained by using upper bound constraints leads to more meaningful teams. Moreover, in the cases where the previous density-based method by Gajewar and Das Sarma [2012] was applicable, the CFSP-based method produced teams of higher density and smaller size. We refer to Rangapuram et al. [2013] for a detailed discussion of the results for the team formation problem.

In the following we will focus on the community detection problem with seed and volume constraint and no additional constraints. In the next section, we will derive a tight relaxation of the problem (9.1).

9.2 Tight relaxation of local community detection problem

In this section we will derive a tight relaxation of the local community detection problem (9.1) for general non-negative vertex weights $g, h_1, h_2 \in \mathbb{R}^n_+$. The derivation for this problem is similar to the one for the local clustering problem in Chapter 8.

9.2.1 Elimination of volume constraints

First, we integrate the volume constraint via a penalty term which yields the following equivalent problem. We use again the set function $\widehat{P}$ given by $\widehat{P}(\emptyset) = 0$, and $\widehat{P}(C) = 1$, if $C \neq \emptyset$.

Lemma 9.1. *The problem* (9.1) *is equivalent to the problem*

$$\begin{aligned} &\min_{C \subset V} \frac{\mathrm{vol}_g(C) + \gamma \widehat{T}_{k_1,k_2}(C)}{\mathrm{assoc}(C)}, \\ &\text{subject to} : J \subset C \end{aligned} \tag{9.2}$$

where

$$\widehat{T}_{k_1,k_2}(C) = k_1 \widehat{P}(C) + \mathrm{vol}_{h_2}(C) - \min\left\{k_1, \mathrm{vol}_{h_1}(C)\right\} - \min\left\{k_2, \mathrm{vol}_{h_2}(C)\right\}$$

and $\gamma > \frac{\mathrm{vol}_g(C_0)\,\mathrm{vol}(V)}{\theta\,\mathrm{assoc}(C_0)}$ *for some feasible set* $C_0 \subset V$*, and* $\theta > 0$ *is a constant.*

Proof. First one rewrites the maximization problem in (9.1) as a minimization problem by exchanging numerator and denominator. We now derive penalty terms for the constraints $k_1 \leq \mathrm{vol}_{h_1}(C)$ and $\mathrm{vol}_{h_2}(C) \leq k_2$ according to Eq. (4.6). The lower bound constraint is first rewritten as an upper bound constraint $-\mathrm{vol}_{h_1}(C) \leq -k1$, which yields the penalty term

$$\begin{aligned} \widehat{T}^{(1)}_{k_1}(C) &= \begin{cases} \max\left\{0, -\mathrm{vol}_{h_1}(C) - (-k1)\right\}, & C \neq \emptyset, \\ 0, & C = \emptyset. \end{cases} \\ &= k_1 \widehat{P}(C) - \min\left\{k_1, \mathrm{vol}_{h_1}(C)\right\}. \end{aligned}$$

For the upper bound constraint we obtain

$$\begin{aligned} \widehat{T}^{(2)}_{k_2}(C) &= \begin{cases} \max\left\{0, \mathrm{vol}_{h_2}(C) - k_2)\right\}, & C \neq \emptyset, \\ 0, & C = \emptyset. \end{cases} \\ &= \mathrm{vol}_{h_2}(C) - \min\left\{k_2, \mathrm{vol}_{h_2}(C)\right\}. \end{aligned}$$

Writing $\widehat{T}_{k_1,k_2}(C) = \widehat{T}^{(1)}_{k_1}(C) + \widehat{T}^{(2)}_{k_2}(C)$, the result then follows by Lemma 4.5 for any γ satisfying

$$\gamma > \frac{\mathrm{vol}_g(C_0)}{\theta\, \mathrm{assoc}(C_0)} \max_{C \subset V} \mathrm{assoc}(C),$$

for some feasible set $C_0 \subset V$, and the constant θ defined in Eq. (4.7). Explicitly, in this case θ is given as

$$\theta = \min \left\{ \min_{\mathrm{vol}_{h_1}(C) > k_1} \left\{ \mathrm{vol}_{h_1}(C) - k_1 \right\}, \min_{\mathrm{vol}_{h_2}(C) < k_2} \left\{ k_2 - \mathrm{vol}_{h_2}(C) \right\} \right\}.$$

Noting that $\max_{C \subset V} \mathrm{assoc}(C) = \mathrm{vol}(V)$, we obtain the result. □

Similarly to the constrained normalized cut problem from Chapter 8, we discuss two different ways to handle the subset constraint. The first approach reformulates the problem as an optimization problem on the graph obtained after excluding the seed subset, the second one handles the subset constraint via a penalty term.

9.2.2 Direct integration of seed subset

In this section we show how the subset constraint $J \subset C$ can be incorporated directly into the objective. By writing the set C as $C = A \cup J$ for some set $A \subset V$ with $A \cap J = \emptyset$ one observes that the problem boils down to finding the optimal set A and we can rewrite the problem as an optimization problem on the graph with vertices $V' = V \backslash J$. An illustration is given in Fig. 8.2.

Lemma 9.2. *The problem* (9.2) *is equivalent to the problem*

$$\min_{A \subset V \backslash J} \frac{\mathrm{vol}_g(A) + \mathrm{vol}_g(J) + \gamma(\widehat{T}_{k'_1,k'_2}(A) + k'_1(1 - \widehat{P}(A)))}{\mathrm{vol}_d(A) + \mathrm{assoc}(J) + \mathrm{cut}(J, A) - \mathrm{cut}(A, V' \backslash A)}, \tag{9.3}$$

where $k'_1 = k_1 - \mathrm{vol}_{h_1}(J)$ *and* $k'_2 = k_2 - \mathrm{vol}_{h_2}(J)$ *and*

$$\gamma > \frac{(\mathrm{vol}_g(A_0) + \mathrm{vol}_g(J))\, \mathrm{vol}_d(V)}{\theta\, (\mathrm{vol}_d(A_0) + \mathrm{assoc}(J) + \mathrm{cut}(J, A_0) - \mathrm{cut}(A_0, V' \backslash A_0))}$$

for some feasible set $A_0 \subset V'$. *Solutions* C^* *of* (9.2) *and* A^* *of* (9.3) *are related via* $C^* = A^* \cup J$.

Proof. Writing $C = A \cup J$, where $A \subset V$ with $A \cap J = \emptyset$, and introducing the notation $V' = V \backslash J$, we decompose the terms in (9.2) as follows:

$$\mathrm{vol}_g(C) = \sum_{i \in C} g_i = \sum_{i \in A} g_i + \sum_{i \in J} g_i = \mathrm{vol}_g(A) + \mathrm{vol}_g(J),$$

$$\operatorname{assoc}(C) = \sum_{i,j\in C} w_{ij} = \sum_{i,j\in A} w_{ij} + \sum_{i,j\in J} w_{ij} + 2\sum_{i\in A, j\in J} w_{ij},$$

where we used the symmetry of W. Using that $V = (V\backslash(A\cup J))\cup A\cup J = (V'\backslash A)\cup A\cup J$, this can be rewritten as

$$\begin{aligned}&\sum_{i\in A, j\in V} w_{ij} - \sum_{i\in A, j\in V'\backslash A} w_{ij} - \sum_{i\in A, j\in J} w_{ij} + \sum_{i,j\in J} w_{ij} + 2\sum_{i\in A, j\in J} w_{ij}\\ &= \operatorname{vol}_d(A) + \operatorname{assoc}(J) + \operatorname{cut}(J, A) - \operatorname{cut}(A, V'\backslash A).\end{aligned}$$

Similarly one obtains for the first two terms in the penalty term, $k_1\widehat{P}(C) + \operatorname{vol}_{h_2}(C) = k_1 + \operatorname{vol}_{h_2}(A) + \operatorname{vol}_{h_2}(J)$, where we used the fact that $\widehat{P}(A\cup J) = \widehat{P}(J) = 1$ as we assume that J is non-empty. Moreover,

$$\min\{k_1, \operatorname{vol}_{h_1}(C)\} = \min\{k_1 - \operatorname{vol}_{h_1}(J), \operatorname{vol}_{h_1}(A)\} + \operatorname{vol}_{h_1}(J),$$

and analogously for the fourth term in $\widehat{T}_{k_1,k_2}$. Thus in total we get

$$\begin{aligned}\widehat{T}_{k_1,k_2}(C) =& k_1 - \operatorname{vol}_{h_1}(J) + \operatorname{vol}_{h_2}(A) - \min\{k_1 - \operatorname{vol}_{h_1}(J), \operatorname{vol}_{h_1}(A)\}\\ &- \min\{k_2 - \operatorname{vol}_{h_2}(J), \operatorname{vol}_{h_2}(A)\}.\end{aligned}$$

Introducing $k_1' = k_1 - \operatorname{vol}_{h_1}(J)$ and $k_2' = k_2 - \operatorname{vol}_{h_2}(J)$, we can write this as

$$\widehat{T}_{k_1,k_2}(C) = \widehat{T}_{k_1',k_2'}(A) + k_1'(1 - \widehat{P}(A)) = \begin{cases}\widehat{T}_{k_1',k_2'}(A), & \text{if } A \neq \emptyset\\ k_1', & \text{if } A = \emptyset\end{cases}.$$

Replacing the terms in (9.2) gives the result. Thus a solution C^* of (9.2) is obtained by computing a solution A^* of (9.3) and setting $C^* = A^*\cup J$. □

Similar to the derivation for the constrained NCut problem in Chapter 8, we need to do a small modification. As the Lovász extension of a set function $\widehat{S}$ in Def. 2.12 requires the function to satisfy $\widehat{S}(\emptyset) = 0$, we replace the constant set functions $\operatorname{vol}_g(J)$, $\operatorname{assoc}(J)$ and k_1' by $\operatorname{vol}_g(J)\widehat{P}(A)$, $\operatorname{assoc}(J)\widehat{P}(A)$ and $k_1'\widehat{P}(A)$, respectively. This leads to the problem

$$\min_{A\subset V\backslash J} \frac{\operatorname{vol}_g(A) + \operatorname{vol}_g(J)\widehat{P}(A) + \gamma\widehat{T}_{k_1',k_2'}(A)}{\operatorname{vol}_d(A) + \operatorname{assoc}(J)\widehat{P}(A) + \operatorname{cut}(J, A) - \operatorname{cut}(A, V'\backslash A)}. \tag{9.4}$$

The only difference to (9.2) lies in the treatment of the empty set. Note that with $\frac{0}{0} := \infty$ the empty set can never be optimal for (9.4). Given an optimal solution A^* of (9.4), one then considers either $A^*\cup J$ or J, depending on whichever has lower objective, which then implies equivalence to (9.2).

The resulting tight relaxation will be a minimization problem over $\mathbb{R}^m$ with $m = |V\backslash J|$ and we assume wlog that the first m vertices of V are the ones in $V\backslash J$. Moreover, we use the notation $f_{\max} = \max_{i=1,\dots,m} f_i$ for

$f \in \mathbb{R}^m$, and $d_i^{(A)} = \sum_{j \in A} w_{ij}$. In order to derive the tight relaxation, we need the Lovász extensions of the set functions in (9.4). In Table 9.1, we recollect some Lovász extensions already encountered earlier which will be used in the derivation of the tight relaxation presented in Theorem 9.3, as well as later in Theorem 9.5. The proofs can be found in Prop. 2.22 and Prop. 2.23 as well as Lemmas 8.4, 8.5, 8.6 and 8.9.

Table 9.1: Lovász extensions used in tight relaxation of general maximum density subgraph problem.

Set function	Lovász extension	Shown in
$\mathrm{vol}_g(A)$	$\langle f, (g_i)_{i=1}^m \rangle$	Prop. 2.22
$\mathrm{cut}(A, V' \backslash A)$	$\frac{1}{2} \sum_{i,j \in V'} w_{ij} \lvert f_i - f_j \rvert$	Prop. 2.23
$\widehat{P}(A)$	$f_{\max}$	Lemma 8.4
$\mathrm{cut}(J, A)$	$\langle d^{(J)}, f \rangle$	Lemma 8.5
$\min\{k, \mathrm{vol}_g(A)\}$	$\langle f, t_{k'}^{(2)}(f) \rangle$	Lemma 8.6
$\lvert C \cap J \rvert$	$\langle \mathbf{1}_J, f \rangle$	Lemma 8.9

Theorem 9.3 (Tight relaxation of maximum density problem). *The problem in* (9.4) *is equivalent to the problem*

$$\min_{f \in \mathbb{R}_+^m} \frac{R_1(f) - R_2(f)}{S_1(f) - S_2(f)}, \tag{9.5}$$

where the convex functions R_1, R_2, S_1, S_2 are given as

$$\begin{aligned}
R_1(f) &= \langle f, (g_i)_{i=1}^m + \gamma((h_2)_i)_{i=1}^m \rangle \; + \; \left(\mathrm{vol}_g(J) + \gamma k_1'\right) f_{\max}, \\
R_2(f) &= \gamma \left\langle f, t_{k_1'}^{(2)}(f) + t_{k_2'}^{(2)}(f) \right\rangle, \\
S_1(f) &= \langle f, (d_i)_{i=1}^m + (d_i^{(J)})_{i=1}^m \rangle \; + \; \mathrm{assoc}(J) f_{\max}, \quad \textit{and} \\
S_2(f) &= \tfrac{1}{2} \textstyle\sum_{i,j \in V'} w_{ij} \lvert f_i - f_j \rvert.
\end{aligned}$$

Proof. The objective in (9.4) can be written as a ratio of two differences of submodular set functions as follows

$$\min_{A \subset V'} \frac{\widehat{R}_1(A) - \widehat{R}_2(A)}{\widehat{S}_1(A) - \widehat{S}_2(A)},$$

where the submodular functions $\widehat{R}_1$, $\widehat{R}_2$, $\widehat{S}_1$, $\widehat{S}_2$ are given as

$$\begin{aligned}
\widehat{R}_1(A) &= \mathrm{vol}_g(A) + \gamma \, \mathrm{vol}_{h_2}(A) + \mathrm{vol}_g(J) \widehat{P}(A) + \gamma k_1' \widehat{P}(A), \\
\widehat{R}_2(A) &= \gamma \min\{k_1', \mathrm{vol}_{h_1}(A)\} + \gamma \min\{k_2', \mathrm{vol}_{h_2}(A)\}, \\
\widehat{S}_1(A) &= \mathrm{vol}_d(A) + \mathrm{cut}(J, A) + \mathrm{assoc}(J) \widehat{P}(A), \quad \text{and} \\
\widehat{S}_2(A) &= \mathrm{cut}(A, V' \backslash A).
\end{aligned}$$

One now replaces the set functions by their Lovász extensions, see Table 9.1. This directly leads to the terms R_1, R_2, S_1 and S_2, which by Theorem 4.4 yield a tight relaxation of the original problem. □

9.2.3 Seed constraint via penalty function

Here we briefly state the results for the alternative approach where the subset constraint is incorporated into the problem via a penalty function.

Lemma 9.4. *The problem* (9.1) *is equivalent to the problem*

$$\begin{aligned} \min_{C \subset V} \ & \frac{\mathrm{vol}_g(C) + \gamma \left(|J| - |C \cap J| \right)}{\mathrm{vol}(C) - \mathrm{cut}(C, \overline{C})} \\ \text{subject to} : \ & k_1 \leq \mathrm{vol}_{h_1}(C), \\ & \mathrm{vol}_{h_2}(C) \leq k_2, \end{aligned} \tag{9.6}$$

where $\gamma > \frac{\mathrm{vol}_g(C_0)\,\mathrm{vol}(V)}{\theta\,\mathrm{assoc}(C_0)}$ *for some feasible set* $C_0 \subset V$.

Proof. We first turn the maximization problem into a minimization problem by exchanging numerator and denominator. Moreover, note that

$$\mathrm{assoc}(C) = \sum_{i,j \in C} w_{ij} = \sum_{i \in C, j \in V} w_{ij} - \sum_{i \in C, j \in \overline{C}} w_{ij} = \mathrm{vol}(C) - \mathrm{cut}(C, \overline{C}).$$

The proof then works analogously to the proof of Lemma 8.8. □

In the following, the volume constraints are omitted for clarity of presentation. Moreover, we can replace the constant function $|J|$ by the function $|J|\,\widehat{P}(C)$, see the discussion after Lemma 9.2. This leads to the problem

$$\min_{C \subset V} \ \frac{\mathrm{vol}_g(C) + \gamma \big(|J|\,\widehat{P}(C) - |C \cap J| \big)}{\mathrm{vol}(C) - \mathrm{cut}(C, \overline{C})}. \tag{9.7}$$

Theorem 9.5 (Tight relaxation with subset as penalty). *The problem in* (9.7) *is equivalent to the problem*

$$\min_{f \in \mathbb{R}^m_+} \ \frac{R_1(f) - R_2(f)}{S_1(f) - S_2(f)}, \tag{9.8}$$

where the convex functions R_1, R_2, S_1, S_2 *are given as*

$$\begin{aligned} R_1(f) &= \big\langle f, g \big\rangle + \gamma\,|J|\,f_{\max}, \\ R_2(f) &= \gamma \big\langle f, \mathbf{1}_J \big\rangle, \\ S_1(f) &= \big\langle f, d \big\rangle \quad \textit{and} \\ S_2(f) &= \tfrac{1}{2} \textstyle\sum_{i,j \in V} \big| f_i - f_j \big|. \end{aligned}$$

Proof. The objective in (9.7) can be written as a ratio of two differences of submodular set functions as follows

$$\min_{C \subset V} \frac{\widehat{R}_1(C) - \widehat{R}_2(C)}{\widehat{S}_1(C) - \widehat{S}_2(C)},$$

where the submodular functions $\widehat{R}_1$, $\widehat{R}_2$, $\widehat{S}_1$, $\widehat{S}_2$ are given as

$$\begin{aligned} \widehat{R}_1(C) &= \operatorname{vol}_g(C) + \gamma\,|J|\,\widehat{P}(C), \\ \widehat{R}_2(C) &= \gamma\,|J \cap C|\,, \\ \widehat{S}_1(C) &= \operatorname{vol}(C), \quad \text{and} \\ \widehat{S}_2(C) &= \operatorname{cut}(C, \overline{C}). \end{aligned}$$

Replacing the set functions by their Lovász extensions, see Table 9.1, then yields the result via Theorem 4.4. □

9.3 Solution via RatioDCA

Observe that both numerator and denominator of the tight relaxation (9.5) and (9.8) are 1-homogeneous d.c. functions and thus we can apply the RatioDCA of Section 5. The crucial step in the algorithm is solving the inner problem. It turns out that in both cases it has the form

$$\min_{\substack{f \in \mathbb{R}^m_+ \\ \|f\|_2 \leq 1}} \left\{ \tfrac{\lambda^k}{2} \sum_{i,j}^{m} w_{ij} \big| f_i - f_j \big| + c_1 f_{\max} + \langle f, c_2^k \rangle \right\},$$

where $c_1 \in \mathbb{R}$ is a constant and c_2^k is a vector depending on the current iterate f^k. Thus, up to the factor λ^k it has the same structure as the inner problem for the constrained local clustering problems in Chapter 8. The explicit values of c_1 and c_2^k are given for the problem in (9.5) as

$$\begin{aligned} c_1 &= \operatorname{vol}_g(J) + \gamma k_1' - \lambda^k \operatorname{assoc}(J), \\ c_2^k &= (g_i)_{i=1}^m + \gamma((h_2)_i)_{i=1}^m - \gamma\Big(t^{(2)}_{k_1'}(f) + t^{(2)}_{k_2'}(f)\Big) - \lambda^k\Big((d_i)_{i=1}^m + (d_i^{(J)})_{i=1}^m\Big), \end{aligned} \tag{9.9}$$

and for the problem in (9.8) as

$$\begin{aligned} c_1 &= \gamma\,|J|\,, \\ c_2^k &= g - \gamma \mathbf{1}_J - \lambda^k d. \end{aligned} \tag{9.10}$$

Dividing by $\lambda^k > 0$ yields an inner problem of the same structure as for the local clustering problem, thus we can solve this problem analogously via its dual problem, see Section 8.3. The algorithm is summarized in Alg. 20.

20 Algorithm for local community detection

Input: weight matrix W
Initialization: nonconstant f^0 with $\left\|f^0\right\|_2 = 1$, accuracy ϵ
repeat
  Compute c_1 and c^k according to (9.9) for (9.5) or (9.10) for (9.8).
  $f^{k+1} = \underset{\substack{f\in\mathbb{R}^m_+ \ / \ f\in\mathbb{R}^n_+ \\ \|f\|_2\leq 1}}{\arg\min} \left\{ \frac{\lambda^k}{2} \sum_{i,j}^m w_{ij}|f_i - f_j| + c_1 f_{\max} + \left\langle f, c_2^k \right\rangle \right\}$,
  $\lambda^{k+1} = \frac{R_1(f^k) - R_2(f^k)}{S_1(f^k) - S_2(f^k)}$
until $\frac{\left|\lambda^{k+1} - \lambda^k\right|}{\lambda^k} < \epsilon$

Theorem 9.6 (Convergence). *Let Q be the functional in (9.5) or (9.8), depending on the choice of c_1 and c^k. The sequence f^k produced by Alg. 20 satisfies $Q(f^k) > Q(f^{k+1})$ for all $k \geq 0$ or terminates. Moreover, f^k has a subsequence converging to a solution of the eigenproblem associated to Q.*

Proof. This is a direct corollary of Prop. 5.9 and Theorem 5.13. □

We omit the exact form of the eigenproblem associated to the problems in (9.5) or (9.8). Similar to the previous applications from Chapter 7 and 8, one can give the following improvement guarantee.

Theorem 9.7 (Density improvement). *Let C be any feasible set, f denote the result of Alg. 20 after initializing with $\frac{1}{|C|}\mathbf{1}_C$, and C_f be the set obtained by optimal thresholding of f. Either Alg. 20 terminates after one iteration, or the set C_f is feasible and it holds that* $\text{Density}_g(C_f) > \text{Density}_g(C)$.

Proof. This is a direct corollary of Theorem 5.13. □

Next we consider the solution of the inner problem in the special case of the unconstrained maximum density subgraph problem.

9.3.1 Unconstrained version

In the unconstrained case of the maximum density problem, the tight relaxation (9.8) reduces to a convex-concave ratio, given as

$$\min_{f\in\mathbb{R}^n_+} \frac{\left\langle f, g \right\rangle}{\left\langle f, d \right\rangle - \frac{1}{2}\sum_{i,j\in V} \left|f_i - f_j\right|}. \tag{9.11}$$

Due to the 1-homogeneity of the objective, we can replace the constraint $f \in \mathbb{R}^n_+$ by the constraint $f \in H$, where $H := \left\{ f \in \mathbb{R}^n_+ \mid \|f\|_\infty \leq 1 \right\}$.

By Prop. 2.14, the numerator and denominator of (9.11) are non-negative for all $f \in \mathbb{R}^n_+$, since they are the Lovász extensions of the non-negative set functions $\text{vol}_g(C)$ and $\text{assoc}(C)$, respectively. Thus, the problem can be

solved globally optimally with the method of Dinkelbach [1967] presented in Alg. 2. In every iteration, we have to solve

$$\min_{f \in H} \left\{ \tfrac{\lambda^k}{2} \sum_{i,j=1}^{n} w_{ij} \big| f_i - f_j \big| + \langle f, g - \lambda^k d \rangle \right\}. \tag{9.12}$$

The following proposition shows that (9.12) can be rewritten as a s-t-mincut-problem.

Proposition 9.8. *Problem* (9.12) *is equivalent to the problem*

$$\min_{f_V \in H,\, f_s = 1,\, f_t = 0} \tfrac{1}{2} \sum_{i,j \in V'} w'_{ij} \big| f_i - f_j \big|,$$

with $V' = V \cup \{s,t\}$, $H := \left\{ u \in \mathbb{R}^n_+, \|u\|_\infty \leq 1 \right\}$ *and some non-negative weights* w'_{ij}, $i,j \in V'$.

Proof. Note that adding constant terms to the objective does not change the minimizer. We rewrite the objective as

$$\begin{aligned} &\tfrac{\lambda^k}{2} \sum_{i,j=1}^{n} w_{ij} \left| f_i - f_j \right| + \sum_{i=1}^{n} g_i (f_i - 0) + \lambda^k \sum_{i=1}^{n} d_i - \lambda^k \sum_{i=1}^{n} d_i f_i \\ = &\tfrac{\lambda^k}{2} \sum_{i,j=1}^{n} w_{ij} \left| f_i - f_j \right| + \sum_{i=1}^{n} g_i \left| f_i - 0 \right| + \lambda^k \sum_{i=1}^{n} d_i \left| 1 - f_i \right|, \end{aligned}$$

where we have used that every minimizer $f \in \mathbb{R}^n_+$ has $\|f\|_\infty \leq 1$, i.e. $f_i \in [0,1], \forall i = 1, \ldots, n$. We define the graph as $V' = V \cup \{s,t\}$ and the weight matrix W' with

$$w'_{ij} = \begin{cases} \lambda\, w_{ij} & \text{if } i,j \in V, \\ 2\, \lambda^k\, d_j & \text{if } i = s \text{ and } j \in V, \\ 2\, g_i & \text{if } i \in V \text{ and } j = t, \end{cases}$$

and rewrite the problem as

$$\min_{f_V \in H,\, f_s = 1,\, f_t = 0} \tfrac{1}{2} \sum_{i,j \in V'} w'_{ij} \big| f_i - f_j \big|.$$

By Prop. 2.19 as well as Prop. 2.23 and Lemma 8.5, this is equivalent to

$$\min_{C \subset V} \operatorname{cut}(C, \overline{C}) + \operatorname{cut}(\{s\}, \overline{C}) + \operatorname{cut}(\{t\}, C),$$

which is a s-t-mincut. □

As discussed in Section 7, the s-t-mincut problem can be efficiently solved, e.g. using the pseudo-flow algorithm of Hochbaum [2008].

9.4 Experimental results

In all experiments we start the RatioDCA with 10 different random initializations and report the result with smallest objective value. As in the case of the local clustering method from Chapter 8, the parameter γ from Theorem 4.6, is obtained by first solving the unconstrained case ($\gamma = 0$) and then increasing γ sequentially, until all constraints are fulfilled.

9.4.1 Community detection on DBLP data

We evaluate our approach for local community detection on a co-author network constructed from the DBLP publication database. Each node in the network represents a researcher and an edge between two nodes indicates a common publication. The task is to extract communities of researchers around a given seed set.

The weights of the graph are defined as $w_{ij} = \sum_{l \in P_i \cap P_j} \frac{1}{|A_l|}$, where P_i, P_j denotes the set of publications of authors i and j and A_l denote the sets of authors for publication l, i.e. the weights represent the total contribution to shared papers. This normalization avoids the problem of giving high weight to a researcher who has publications that have a large number of authors, which usually does not reflect close collaboration with all co-authors. To avoid finding a trivial densely connected group of researchers with only few connections to the rest of the authors, we restrict the graph by considering only authors with at least two publications and maximum distance two from the seed set. As volume function vol_g in (9.1), we use the volume of the original graph in order to further enforce densely connected components.

We perform local community detection with the size constraint $|C| \leq 20$ and three different seed sets

$$
\begin{aligned}
J_1 &= \{\text{P. Bartlett, P. Long, G. Lugosi}\}, \\
J_2 &= \{\text{E. Candes, J. Tropp}\} \text{ and} \\
J_3 &= \{\text{O. Bousquet}\}.
\end{aligned}
$$

The results are shown in Fig. 9.2. The seed set J_1 consists of well-known researchers in learning theory, and all members of the detected community work in this area. To validate this, we counted the number of publications in the two main theory conferences COLT and ALT. At the time of the first publication of this result, each author on average had 18.2 publications in these two conferences (see Table 9.2 for more details). The seeds J_2 yield a community of key scientists in the field of sparsity such as T. Tao, R. Baraniuk, J. Romberg, M. Wakin and R. Vershynin. The third community contains researchers who are or were members of the group of B. Schölkopf or have closely collaborated with his group.

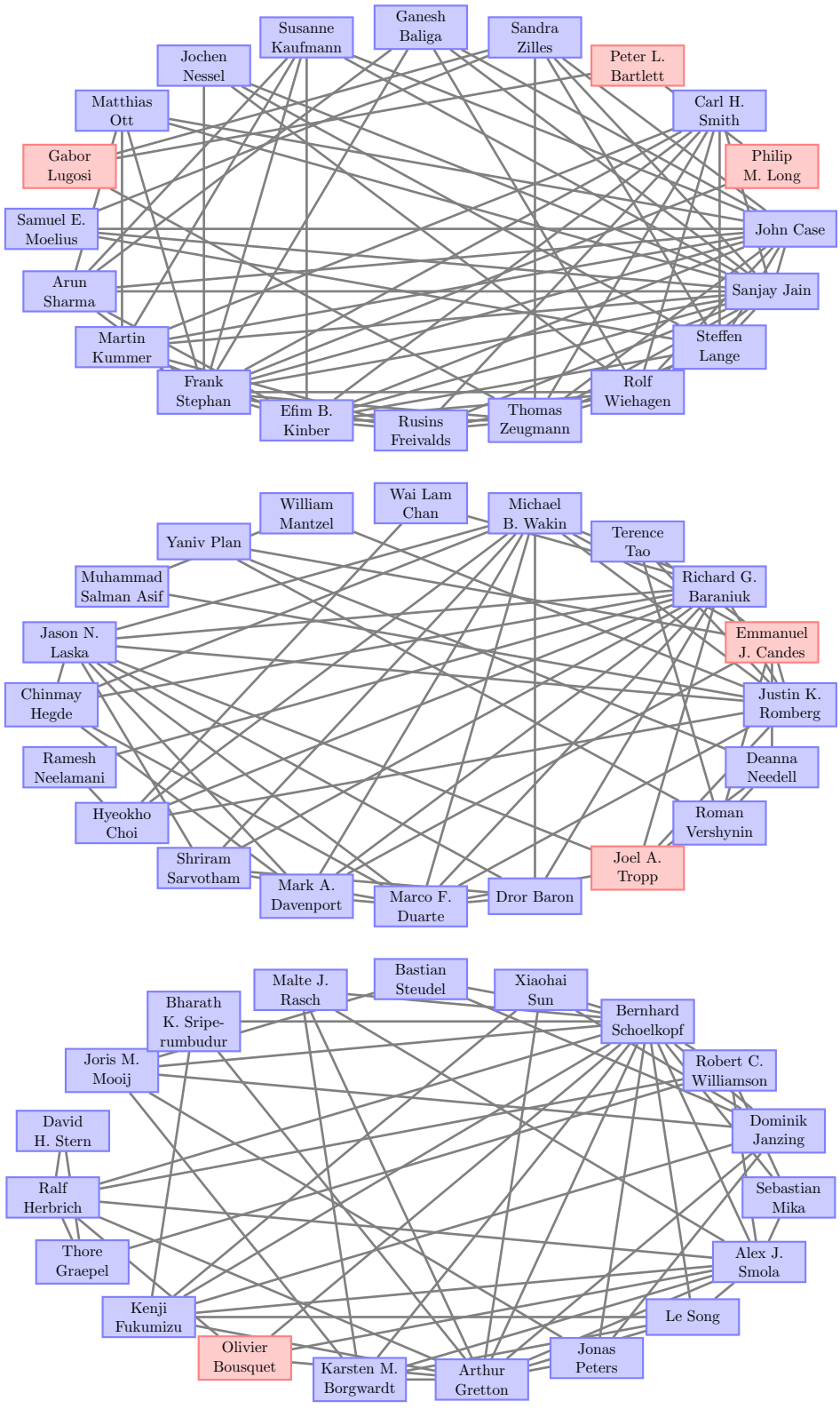

Figure 9.2: Results for community detection on DBLP co-author network. *Top:* Learning theory *Middle:* Sparsity *Bottom:* Kernels

Table 9.2: The number of publications in ALT and COLT of each author in the "learning theory" community.

Author	COLT	ALT
Sandra Zilles	3	13
Peter L. Bartlett	24	2
Carl H. Smith	13	4
Philip M. Long	21	3
John Case	12	18
Sanjay Jain	21	40
Steffen Lange	14	5
Rolf Wiehagen	6	7
Thomas Zeugmann	6	20
Rusins Freivalds	6	5
Efim B. Kinber	11	9
Frank Stephan	13	28
Martin Kummer	5	0
Arun Sharma	10	13
Samuel E. Moelius	1	5
Gabor Lugosi	16	1
Matthias Ott	2	1
Jochen Nessel	1	2
Susanne Kaufmann	1	1
Ganesh Baliga	1	0

9.4.2 Community detection on composer network

In this experiment we evaluate our approach to community detection on a network of classical composers. A subset of the Amazon product catalog was downloaded from [Leskovec] containing metadata for Amazon products available in 2006. For each item, the data contains a list of similar items, based on co-purchase with the given item.

Given a network of composers where the edges represent some notion of similarity, detecting communities is a useful technique which can be used in the context of a recommendation engine. The idea is that a user who likes music by one composer will likely be interested in compositions by a similar artist. Thus we apply our community detection algorithm on the composer-composer network, with the aim to find meaningful communities of similar composers in the network (note that here we use a wider definition of the term "community"). Moreover, note that while in this experiment we restrict ourselves to the subset of classical music records, the same technique could be applied for records of other genres, or other types of networks such as a network between book authors or movie directors.

First we use the available data to construct a network of classical composers as follows: two composers are connected to each other if one or more

recordings of their works were co-purchased together. Given the matrix $W \in \{0,1\}^{m \times m}$ of record similarity based on co-purchases, i.e. $W_{ij} = 1$ if record j is among the similar items of record i (note that this graph is not symmetric), a matrix $A \in \{0,1\}^{n \times m}$ containing the mapping from records to composers as well as a diagonal matrix $D \in \mathbb{N}^{n \times n}$ containing the number of records for each composer, i.e. $D_{ii} = \sum_{j=1}^{m} A_{ij}$, the weight matrix $W_{comp} \in \mathbb{R}^{n \times n}$ for the composer-composer graph is constructed as

$$W_{comp} = D^{-1} A \overline{W} A^T D^{-1}, \qquad \text{where } \overline{W} = 0.5 \cdot \left(W + W^T\right) .$$

We then eliminate self-edges and consider the largest connected component of the resulting graph. Furthermore, note that we restrict ourselves to records having only one composer in order to filter out compilation albums.

The use of the matrix D limits the impact of very popular composers where a very large number of records exist. This is a desirable property for a recommendation system, since usually the popularity of items can be described by a power law distribution, where most of the items are contained in the heavy tail of the distribution, i.e. there are only relatively few items which are very popular, while most of the items are sold very infrequently. Thus, a recommendation system should make the user aware of these items which are unknown but may be similar to the user's taste [Anderson, 2006].

We perform local community detection with the size constraint $|C| \leq 10$ and the four different seeds

$$\begin{aligned} J_1 &= \{\text{Byrd, William}\}, \\ J_2 &= \{\text{Bach, Johann Sebastian}\}, \\ J_3 &= \{\text{Schoenberg, Arnold}\} \text{ and} \\ J_4 &= \{\text{Cage, John}\}. \end{aligned}$$

Table 9.3 shows the communities found by our algorithm. We are able to retrieve different communities of composers from different musical eras. To illustrate this, we give the times of birth and death as well as the corresponding musical period for each composer [Sadie and Grove, 2001]. Note that in some cases the exact dates are not known and estimates are given.

The community found with query J_1 consists of composers mainly from the renaissance and early baroque period as well as composers/lyricists from the medieval era. This community can be roughly identified with the musical era "early music". The seed set J_2 mainly yields composers of the baroque era. The example obtained with query J_3 is a very dense community containing the core members of what is commonly referred to as the "Second Viennese School", a group of influential composers consisting mainly of Arnold Schoenberg and his pupils Alban Berg and Anton von Webern [Sadie and Grove, 2001]. Finally, the community obtained with query J_4 contains well-known 20th century composers.

Table 9.3: Results of community detection on network of classical composers (seed in bold). The found communities can be roughly identified with musical eras 'Early music', 'Baroque', 'Modern' and '20th century'.

Composer	Birth	Death	Period
Byrd, William	**1540**	**1623**	**Renaissance**
Dufay, Guillaume	1397	1474	Renaissance
Gombert, Nicolas	1495	1560	Renaissance
Kapsberger, Giovanni	1580	1651	Baroque
Lassus, Orlande de	1532	1594	Renaissance
Lobo, Alonso	1555	1617	Renaissance
Machaut, Guillaume de	1300	1377	Medieval
Obrecht, Jacob	1457	1505	Renaissance
Praetorius, Michael	1571	1621	Renaissance
Walther v. der Vogelweide	1170	1230	Medieval
Bach, Johann Sebastian	**1685**	**1750**	**Baroque**
Biber, Heinrich Ignaz	1644	1704	Baroque
Couperin, Louis	1626	1661	Baroque
Dittersdorf, Karl Ditters	1739	1799	Classical
Pandolfi-Mealli, G.A.	1630	1670	Baroque
Rameau, Jean Philippe	1683	1764	Baroque
Schmelzer, Johann H.	1623	1680	Baroque
Stamitz, Johann Wenzel	1717	1757	Classical
Tartini, Giuseppe	1692	1770	Baroque
Telemann, Georg Philipp	1681	1767	Baroque
Berg, Alban	1885	1935	Modern
Schoenberg, Arnold	**1874**	**1951**	**Modern**
Webern, Anton von	1883	1945	Modern
Boulez, Pierre	1925	–	20th century
Cage, John	**1912**	**1992**	**20th century**
Dutilleux, Henri	1916	2013	20th century
Ligeti, György	1923	2006	20th century
Lutoslawski, Witold	1913	1994	20th century
Nancarrow, Conlon	1912	1997	20th century
Penderecki, Krzysztof	1933	–	20th century
Rzewski, Frederic	1938	–	20th century
Scelsi, Giacinto	1905	1988	20th century
Stockhausen, Karlheinz	1928	2007	20th century

Chapter 10

Sparse PCA

Principal component analysis (PCA) is a standard technique for dimensionality reduction and data analysis [Jolliffe, 2002]. It was independently developed by Pearson [1901] and later Hotelling [1933]. Given a set of observations of a number of (possibly correlated) variables, in PCA a dimensionality reduction is performed via a transformation to a new set of uncorrelated variables such that the first k of these variables explain as much of the variance of the data as possible. Thus, PCA finds the k-dimensional subspace of maximal variance in the data. Solving PCA reduces to a standard linear eigenproblem involving a positive semidefinite symmetric matrix.

As usually all entries of the loading vectors (i.e. the vectors describing the transformation into the new vector space) are nonzero, an interpretation of the principal components is often difficult. This constitutes a disadvantage for instance in the case of gene expression data where one would like the principal components to consist only of a few significant genes, making it easy to be interpreted by a human.

For this reason, in sparse PCA (see e.g. Cadima and Jolliffe [1995], Jolliffe et al. [2003], Zou et al. [2006], Moghaddam et al. [2006], Sriperumbudur et al. [2007], d'Aspremont et al. [2007], d'Aspremont et al. [2008], Sigg and Buhmann [2008], Shen and Huang [2008], Journée et al. [2010]) one enforces sparsity of the loading vectors with the aim of getting a small number of features while at the same time still capturing most of the variance. In other words, one is interested in the optimal trade-off between explained variance and sparsity. In this chapter, we show how sparse PCA can be efficiently solved using our inverse power method from Section 5.3.

10.1 Principal component analysis

Consider a data matrix $X \in \mathbb{R}^{m \times n}$ where each of the m rows represents a point in a n-dimensional space. For instance the points could correspond to m different repetitions of the same experiment and the columns correspond

to n different features which have been measured.

In principal component analysis one is now interested in finding the k-dimensional subspace which explains most of the variance in the data. This has several different applications for instance in data compression, obtaining a good visualization of the data, or finding the underlying structure in the data [Jolliffe, 2002]. Alternatively to the variance interpretation, PCA can be seen as computing the p-dimensional subspace which best approximates the given data in terms of the Euclidean distance, see for example Hastie et al. [2001]. In the following we give a derivation of principal component analysis based on the variance interpretation.

10.1.1 Variance interpretation

Let us first consider the case $p = 1$. Given a $\mathbb{R}^n$-valued multivariate random variable Y, we consider for some direction $f \in \mathbb{R}^n$ with $\|f\|_2 = 1$ the length of the projection of a point $y \in \mathbb{R}^n$ onto the line with direction f, which is given as $\langle f, y\rangle$. For a given f, the variance of the new random variable $\overline{Y} = \langle f, Y\rangle$ is then given as

$$\mathbb{E}\left[\left(\overline{Y} - \mathbb{E}\left[\overline{Y}\right]\right)^2\right] = \mathbb{E}\left[\left(\langle f, Y\rangle - \mathbb{E}\left[\langle f, Y\rangle\right]\right)^2\right] = \mathbb{E}\Big[\langle f, Y - \mathbb{E}\left[Y\right]\rangle^2\Big].$$

Now we assume that the data is a set of observations $x_1, \ldots, x_m$ drawn from the above random distribution. We can write this in compact form as a data matrix $X \in \mathbb{R}^{m\times n}$ containing the samples as row vectors. The sample variance for the direction f is then given as (scaled with factor m)

$$\mathrm{var}_f(X) = \sum_{i=1}^{m} \langle f, x_i - \tfrac{1}{m}\sum_{j=1}^{m} x_j\rangle^2. \tag{10.1}$$

Introducing the (scaled) sample covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$, given as

$$\Sigma = \sum_{i=1}^{m}\Big(x_i - \tfrac{1}{m}\sum_{j=1}^{m} x_j\Big)\Big(x_i - \tfrac{1}{m}\sum_{j=1}^{m} x_j\Big)^T = \left(X - \tfrac{1}{m}X\mathbf{1}\right)^T\left(X - \tfrac{1}{m}X\mathbf{1}\right),$$

we can rewrite (10.1) in compact form as

$$\mathrm{var}_f(X) = f^T\Sigma f. \tag{10.2}$$

Recall that in PCA one is interested in the direction f^* achieving maximal variance. Thus one needs to compute the unit vector with direction f^* which maximizes (10.2). The problem can now be stated as

$$f^* = \operatorname*{arg\,max}_{f\in\mathbb{R}^n,\, \|f\|_2=1} \left\langle f, \Sigma f\right\rangle = \operatorname*{arg\,max}_{f\in\mathbb{R}^n} \frac{\left\langle f, \Sigma f\right\rangle}{\|f\|_2^2}. \tag{10.3}$$

Note that the matrix Σ is symmetric positive semi-definite. Thus by the Rayleigh-Ritz principle (see Chapter 3) the solution of the above problem is given by the eigenvector corresponding to the largest eigenvalue of the covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$.

Let us now consider the general case of $p \geq 1$. Denote by $F \in \mathbb{R}^{n\times p}$ a matrix having as columns p orthonormal vectors. Instead of considering the projection of a point onto the line with direction f, we now consider the projection of the point onto the space spanned by F. Thus the matrix F performs a coordinate transform into a new coordinate system where each column of the matrix F corresponds to a coordinate axis. Given a point $x \in \mathbb{R}^n$, the coordinates of the new point after the coordinate transform are given as $F^T x \in \mathbb{R}^p$. As before, we are now interested in finding coordinate axes F such that there is a high variance in the new coordinates $F^T x$. Intuitively, this means that the new coordinate axes F represent the structure of the data well and could for example be used to obtain a good visualization of the data. As before, we first consider an n-dimensional random variable Y (corresponding to the original data points) and then compute the expected variance of the new k-dimensional random variable $\overline{Y} = F^T Y$ (corresponding to the projected points). We obtain

$$\mathbb{E}\left[\left(\overline{Y} - \mathbb{E}\left[\overline{Y}\right]\right)^T\left(\overline{Y} - \mathbb{E}\left[\overline{Y}\right]\right)\right] = \mathbb{E}\left[\left\|F^T\left(Y - \mathbb{E}\left[Y\right]\right)\right\|_2^2\right].$$

The generalization of (10.1) then becomes (note that again we omitted the constant factor)

$$\mathrm{var}_F(X) = \sum_{i=1}^{m}\left\|F^T(x_i - \frac{1}{m}\sum_{j=1}^{m} x_j)\right\|_2^2. \tag{10.4}$$

The expression in (10.4) can be rewritten as follows.

Lemma 10.1. *It holds that* $\mathrm{var}_F(X) = \mathrm{tr}(F^T\Sigma F)$.

Proof. We introduce the notation $\tilde{x}_i = x_i - \frac{1}{m}\sum_{i=1}^{m} x_j$, as well as $\widetilde{X} := X - \frac{1}{m}X\mathbf{1}$. Then we can rewrite $\mathrm{var}_F(X)$ as follows:

$$\mathrm{var}_F(X) = \sum_{i=1}^{m}\tilde{x}_i^T F F^T \tilde{x}_i = \sum_{i=1}^{m}(\widetilde{X}FF^T\widetilde{X}^T)_{ii} = \sum_{i=1}^{m}\sum_{j=1}^{n}(\widetilde{X}F)_{ij}(F^T\widetilde{X}^T)_{ji}$$
$$= \sum_{j=1}^{n}\sum_{i=1}^{m}(F^T\widetilde{X}^T)_{ji}(\widetilde{X}F)_{ij} = \sum_{j=1}^{n}(F^T\widetilde{X}^T\widetilde{X}F)_{jj} = \mathrm{tr}(F^T\Sigma F).$$

□

Thus, the PCA problem for $p \geq 1$ can be stated as

$$\begin{aligned} \max \quad & \mathrm{tr}(F^T\Sigma F) \\ \text{subject to} \quad & F^T F = I. \end{aligned} \tag{10.5}$$

The maximum is achieved by the matrix $V \in \mathbb{R}^{n\times p}$ having as columns the eigenvectors corresponding to the p largest eigenvalues $\lambda_n \geq \cdots \geq \lambda_{n-p+1}$, see for example Lütkepohl [1997]. The total variance in this p-dimensional subspace is then given by

$$\mathrm{var}_V(X) = \mathrm{tr}(V^T \Sigma V) = \sum_{i=n-p+1}^{n} v_i^T \Sigma v_i = \sum_{i=n-p+1}^{n} \lambda_i.$$

Note that in the literature both the column vectors of V as well as the columns of the new variables $Z = XV$ are sometimes referred to as *principal components* of Σ. To avoid confusion, we follow the nomenclature in Jolliffe [2002] and refer to the columns of Z as the principal components, while the columns of V are the corresponding *loading vectors*. Formally, one can give the following definition.

Definition 10.2 (Principal components). *Let $X \in \mathbb{R}^{m\times n}$ be a data matrix and the sample covariance matrix be given by $\Sigma = \widetilde{X}^T \widetilde{X}$, where $\widetilde{X} := X - \frac{1}{m} X\mathbf{1}$. Let the matrix $V \in \mathbb{R}^{n\times p}$ contain as columns the eigenvectors corresponding to the p largest eigenvalues of Σ. Let the matrix $Z \in \mathbb{R}^{m\times p}$ be obtained by the linear transformation $Z = \widetilde{X}V$. Then for $k \in \{1, \ldots, p\}$, the k-th column of Z is called the kth principal component (PC) of Σ, and the corresponding column of V is called loading vector for the kth PC.*

In Fig. 10.1 we perform principal component analysis on some random data sampled from a Gaussian distribution in $\mathbb{R}^2$. In the left plot, we give the original data, and the red lines correspond to the eigenvectors of the covariance matrix Σ. On the right, we show the same data with respect to the new coordinate axes Z. In the new coordinate system, the data has a simpler structure and thus becomes easier to analyze.

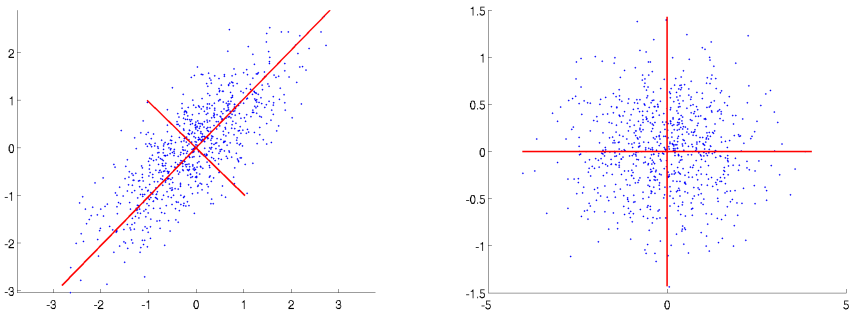

Figure 10.1: *Left:* The loading vectors for the first two principal components for some data drawn from a multivariate Gaussian distribution in $\mathbb{R}^2$. *Right:* After variable transformation to the new coordinate axes.

Note that the term $\widetilde{X} := X - \frac{1}{m}X\mathbf{1}$ appearing in Σ denotes the centering of the points $x_1, \ldots, x_m$, where the mean has been subtracted for each point. Thus, without loss of generality, we can from now on assume that the data is centered, i.e. each column in X has mean 0. Hence the sample covariance matrix can be written compactly as $\Sigma = X^T X$.

In the following, we will show the connection to singular value decomposition (SVD).

10.1.2 Connection to singular value decomposition

To compute the k largest eigenvectors of the covariance matrix, and hence the k-dimensional subspace explaining most of the variance in the data, one makes use of the connection of PCA to singular decomposition of the data matrix (SVD). Singular value decomposition is a standard factorization of a matrix in linear algebra, see e.g. Golub and Van Loan [1996]. Let us first give the following definition.

Definition 10.3 (Singular values). *Let $M \in \mathbb{R}^{m\times n}$. The non-negative number σ is called a singular value of M if there exist unit-length vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$ such that $Mv = \sigma u$ and $M^T u = \sigma v$. The vectors u and v are called left- and right-singular vectors, respectively.*

Given a matrix $M \in \mathbb{R}^{m\times n}$, then there always exists [Lütkepohl, 1997] a decomposition of M into three matrices $M = U\Gamma V^T$, where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthonormal matrices, i.e. $U^T U = 1$ and $V^T V = 1$, and $\Gamma \in \mathbb{R}^{m\times n}$ is a diagonal matrix, where the non-negative entries σ_i are the singular values of M. Moreover, the column vectors of U are the left-singular vectors and the column vectors of V the right-singular vectors of M.

The connection between singular value decomposition and eigendecomposition is now as follows.

Lemma 10.4. *Let $M = U\Gamma V^T$ be the singular value decomposition of M. Then the left-singular vectors U are the eigenvectors of MM^T, and the right-singular vectors V are the eigenvectors of $M^T M$. The corresponding eigenvalues are given by the squares of the corresponding singular values of M.*

Proof. Multiplying the SVD of M from the left with M^T, one obtains $M^T M = M^T U\Gamma V^T = V\Gamma^T U^T U\Gamma V^T = V\Gamma^T\Gamma V$, where we used that $U^T U = I$. Thus we have obtained an eigendecomposition of $M^T M$ with eigenvalues σ_i^2 and eigenvector matrix V. Analogously one proceeds for the second statement by multiplying the matrix from the right with M^T. □

Singular value decomposition has several applications in practice. For instance, in information retrieval, the columns in the matrix $M \in \mathbb{R}^{m\times n}$ may correspond to documents and the rows correspond to terms in the documents, and the entries of the matrix give the relative frequency of the term

in the document. Then, each singular value may correspond to a latent topic in the data, and the matrix U represents a mapping from terms to topics, whereas the matrix V^T represents a mapping from topics to documents [Deerwester et al., 1990]. Another example is given in recommender systems. For example, given a matrix M of ratings of users for a set of movies, the matrix U may define a mapping from users to genres of movies, and the matrix V^T gives a mapping from genres to movies [Sarwar et al., 2000]. In both applications, typically one now would consider only the columns of U and V corresponding to the largest singular values, since these columns correspond to the most significant latent topics/genres in the dataset.

The connection between singular value decomposition and eigendecomposition implies that one can compute the eigenvectors of the covariance matrix $\Sigma = X^T X$ by computing the right singular vectors of the data matrix X. The eigenvalues of Σ are then given by the squares of the non-zero entries of Γ. Moreover, note that the principal components Z of Σ can be written as $Z = XV = U\Gamma V^T V = U\Gamma$.

10.2 Sparse principal component analysis

In order to make sense of some given data, apart from analyzing the factoring of the data obtained by means of the lower-dimensional embedding given by PCA, it is also often useful to interpret the loading vectors of the principal components themselves, as they correspond to the "most meaningful" directions in the given data. Jolliffe [2002] discusses some examples where an interpretation can be found for the principal components obtained by standard PCA: for instance they may correspond to sources of variation for anatomical measurements in different species, different groups in demographic studies, or different types of companies in the analysis of stock market data. However, the datasets considered have only a small number of variables (typically $n \ll 100$). In many applications such as the analysis of gene expression data, the data is very high-dimensional ($n > 10000$). Since in standard PCA usually all entries of the set of eigenvectors of the covariance matrix are nonzero, an interpretation of the principal components becomes difficult in these cases. In order to make an interpretation by a human possible, ideally the loading vectors should consist only of a few significant genes.

For this reason, in *sparse PCA* one enforces sparsity of the loading vectors with the aim of getting a small number of features while at the same time still capturing most of the variance. In other words, one is interested in the optimal trade-off between explained variance and sparsity. In the simplest case for $p = 1$, the problem can be formulated as

$$f^* = \underset{f \in \mathbb{R}^n,\, \|f\|_0 \leq c}{\arg\max} \frac{\langle f, X^T X f \rangle}{\|f\|_2^2}\,, \tag{10.6}$$

where $\|f\|_0$ denotes the number of nonzero components of f, and $c > 0$. It turns out that adding a constraint on the cardinality, i.e. the number of nonzero coefficients, makes the problem NP-hard [Moghaddam et al., 2006].

The first approaches to the problem in (10.6) performed simple thresholding of the principal components, which however was shown to be misleading [Cadima and Jolliffe, 1995]. In the SCoTLASS method [Jolliffe et al., 2003], the cardinality constraint in (10.6) is replaced by a L_1-based constraint, leading to an optimization problem similar to the LASSO [Tibshirani, 1994]. Zou et al. [2006] first reformulate the standard PCA problem as a ridge regression problem. Then they enforce sparsity by adding a L_1 penalty, leading to a elastic net-type problem [Zou and Hastie, 2005]. Sriperumbudur et al. [2007] approximate the cardinality of the loading vectors by a log term and then frame this approximate problem as a d.c. (difference of convex functions) program. Recently, Journée et al. [2010] proposed two single unit (computation of one component only) and two block (simultaneous computation of multiple components) methods based on L_0-penalization and L_1-penalization, which we will discuss in Section 10.3.

A method for the sparse PCA problem called DSPCA is proposed by d'Aspremont et al. [2007] which is based on a relaxation to a semi-definite program. Shen and Huang [2008] use the connection of standard PCA to singular value decomposition (SVD) of the covariance matrix and then compute the sparse principal components using a low-rank matrix approximation problem. Sigg and Buhmann [2008] present an algorithm based on expectation-maximization (EM) for probabilistic PCA [Roweis, 1998, Tipping and Bishop, 1999]. A greedy algorithm is proposed by d'Aspremont et al. [2008] to compute a full set of good candidate solutions up to a specified target sparsity, and sufficient conditions for a vector to be globally optimal are derived. Note that while the problem is NP-hard, Moghaddam et al. [2006] used branch and bound to compute optimal solutions for small problem instances.

10.2.1 Extensions to multiple principle components

So far we have not discussed the question what is a suitable criterion for an extension of sparse PCA to multiple components. As shown in the last section, given a matrix $F = [f_1, \ldots, f_p] \in \mathbb{R}^{n \times p}$ consisting of p orthonormal loading vectors, in standard PCA usually the total variance is calculated as $\mathrm{var}_V(F) = \mathrm{tr}(F^T X^T X F)$. The optimal value is achieved by the matrix $V \in \mathbb{R}^{n \times p}$ consisting of p eigenvectors of the covariance matrix $\Sigma = X^T X$. This criterion makes sense in standard PCA, as the principal components are uncorrelated and the loading vectors are orthogonal.

Several authors proposed extensions of sparse PCA to multiple eigenvectors where the sparse loading vectors are enforced to be orthonormal (as in the case of standard PCA), see e.g. Jolliffe et al. [2003]. The direct

generalization of Eq. (10.5) to the sparse case is given as

$$\begin{aligned} \max \quad & \mathrm{tr}(F^T \Sigma F) \\ \text{subject to} \quad & F^T F = I. \\ & \|f_i\|_0 \leq c, \forall i = 1, \ldots, p, \end{aligned} \tag{10.7}$$

where c is a constant. Thus one enforces sparsity of the loading vectors f_i, while requiring them to be orthonormal.

However, enforcing orthonormality of the sparse loading vectors is somewhat questionable in sparse PCA. Note that in the case of standard PCA, the fact that the loading vectors $(v_i)_{i=1\ldots p}$ are orthonormal as well as a set of eigenvectors of Σ with eigenvalues $(\lambda_i)_{i=1\ldots p}$ implies that $v_i^T \Sigma v_j = \lambda_i \, \delta_{i=j}$, i.e. the components are uncorrelated. Thus, by enforcing orthonormality we have achieved that the data has been transformed in such a way that it is uncorrelated in the new basis. However note that the equivalence between $f_i^T f_j = 0$ and $f_i^T \Sigma f_j = 0$ does not hold in general. For this reason, most recent algorithms for sparse PCA (see e.g. Zou et al. [2006], [d'Aspremont et al., 2007], Shen and Huang [2008], Journée et al. [2010]) do not explicitly enforce orthonormal loading vectors.

Moreover, since in sparse PCA the principal components may be correlated, the criterion $\mathrm{var}_F(X) = \mathrm{tr}(F^T \Sigma F)$ is not a suitable measure to represent the total variance in sparse PCA. This is because for a given l the variance in direction z_l already contains contributions from the previous components $(z_i)_{i=1,\ldots,l-1}$. For this reason, Zou et al. [2006] proposed the *adjusted total variance* as a criterion to evaluate the quality in the presence of correlated components. Given a matrix of principal components Z, the idea is to iteratively adjust the components by only considering the contribution orthogonal to the previous components (note that this is a greedy scheme which depends on the ordering of the principal components). The adjusted variance is then defined as the variance of the adjusted components.

Specifically, given the first principal component $z_1 = X f_1$, one adjusts the second component z_2 by performing a projection orthogonal to z_1, i.e.

$$\hat{z}_2 = z_2 - \tfrac{\hat{z}_1}{\|\hat{z}_1\|_2} (\tfrac{\hat{z}_1}{\|\hat{z}_1\|_2})^T z_2.$$

One now repeats this process iteratively: Given the l-th principal component $z_l = X f_l$, the influence of the previous $l-1$ principal components is eliminated by computing

$$\hat{z}_l = z_l - P_{l-1}(z_l), \quad \text{where } P_{l-1}(z_l) := \sum_{i=1}^{l-1} e_i e_i^T z_l, \quad \text{and } e_i := \tfrac{\hat{z}_i}{\|\hat{z}_i\|_2}. \tag{10.8}$$

Here P_{l-1} denotes the projection on the space spanned by the previous $l-1$ components. The additional variance explained by principal component z_l is then given by $\|\hat{z}_l\|_2^2 = \|X f_l - P_{l-1}(X f_l)\|_2^2$.

Note that the above scheme is exactly the classical Gram-Schmidt procedure to compute a QR decomposition of a matrix Z [Golub and Van Loan, 1996]. To see this, note that we rewrite the equation (10.8) at step l as

$$z_l = \hat{z}_l + \sum_{i=1}^{l-1} e_i e_i^T z_l = \sum_{i=1}^{l} e_i e_i^T z_l,$$

where we used that $e_l = \frac{\hat{z}_l}{\|\hat{z}_l\|_2}$. Formulating this in matrix notation, we observe that we have obtained a QR decomposition $Z = QR$, where Q is the orthonormal matrix having the e_i as columns, and R is an upper triangular matrix with $R_{ij} = e_i^T z_j$, for $i \leq j$. Moreover, one then has $\|\hat{z}_i\|_2^2 = R_{ii}^2$. This motivates the definition of the adjusted variance as [Zou et al., 2006]

$$\text{adjvar}_F(X) := \sum_{i=1}^{k} \|\hat{z}_i\|_2^2 = \sum_{i=1}^{k} R_{ii}^2. \tag{10.9}$$

Note that when the principal components z_i are uncorrelated, the adjusted variance agrees with $\text{var}_F(X) = \text{tr}(F^T \Sigma F)$. We will later derive a deflation scheme to obtain multiple sparse principal components tailored towards the adjusted variance criterion defined above. In the next section, we show how the sparse PCA problem can be modeled as a nonlinear eigenproblem and solved using the inverse power method introduced in this thesis.

10.3 Sparse PCA via nonlinear eigenproblems

In this section we derive a method for sparse PCA based on the nonlinear inverse power method (IPM) introduced in Section 5.3. We first consider the case $p = 1$, where we only compute one sparse loading vector. Problem (10.3) is equivalent to

$$f^* = \arg\min_{f \in \mathbb{R}^n} \frac{\|f\|_2^2}{\langle f, \Sigma f \rangle} = \arg\min_{f \in \mathbb{R}^n} \frac{\|f\|_2}{\|Xf\|_2} \ .$$

In order to enforce sparsity, we replace the L_2-norm by a convex combination of an L_1 norm and L_2 norm in the numerator, which yields the functional

$$F(f) = \frac{(1-\alpha)\|f\|_2 + \alpha \|f\|_1}{\|Xf\|_2} \ , \tag{10.10}$$

with sparsity controlling parameter $\alpha \in [0, 1]$. Standard PCA is recovered for $\alpha = 0$, whereas $\alpha = 1$ yields the sparsest non-trivial solution: the component with the maximal variance. One easily sees that the formulation (10.10) fits in our general framework, as both numerator and denominator are 1-homogeneous convex functions. Thus the ratio problem in (10.10) can be solved efficiently using the nonlinear IPM.

10.3.1 Solution via nonlinear inverse power method

The ratio in (10.10) is a ratio of two convex functions and thus the optimization problem can be solved via the IPM. The convex inner problem of the IPM becomes

$$g^{k+1} = \underset{\|f\|_2 \le 1}{\arg\min} (1-\alpha)\|f\|_2 + \alpha\|f\|_1 - \lambda^k \langle f, \mu^k \rangle \ , \text{ where } \ \mu^k = \frac{\Sigma f^k}{\sqrt{\langle f^k, \Sigma f^k \rangle}}. \tag{10.11}$$

We will show below that this problem has a closed form solution. In the following we use the notation $x_+ = \max\{0, x\}$.

Lemma 10.5. *The convex problem* (10.11) *has the analytical solution*

$$g_i^{k+1} = \tfrac{1}{s}\operatorname{sign}(\mu_i^k)\big(\lambda^k|\mu_i^k| - \alpha\big)_+, \quad \textit{where} \quad s = \sqrt{\textstyle\sum_{i=1}^n \big(\lambda^k|\mu_i^k| - \alpha\big)_+^2} \ .$$

Proof. We note that the objective is positively 1-homogeneous and that the optimum is either zero (achieved by plugging in the previous iterate) or negative, in which case the optimum is attained at the boundary. Thus wlog we can assume that at the optimum $\|f\|_2 = 1$. Thus the problem reduces to

$$\min_{\|f\|_2 \le 1} \alpha\|f\|_1 - \lambda^k \langle f, \mu^k \rangle.$$

First, we derive an equivalent "dual" problem, noting that

$$\alpha\|f\|_1 - \lambda^k \langle \mu^k, f \rangle = \max_{\|v\|_\infty \le 1} \langle f, \alpha v - \lambda^k \mu^k \rangle \ .$$

Using the fact that the objective is convex in f and concave in v and the feasible set is compact, we obtain by the min-max equality:

$$\begin{aligned} \min_{\|f\|_2 \le 1} \max_{\|v\|_\infty \le 1} \langle f, \alpha v - \lambda^k \mu^k \rangle &= \max_{\|v\|_\infty \le 1} \min_{\|f\|_2 \le 1} \langle f, \alpha v - \lambda^k \mu^k \rangle \\ &= \max_{\|v\|_\infty \le 1} -\big\|\alpha v - \lambda^k \mu^k\big\|_2 \ . \end{aligned} \tag{10.12}$$

We observe that the objective of the dual problem is separable in v, as well as the constraints $\|v\|_\infty \le 1$. Thus each component can be optimized separately, which gives

$$v_i^* = \operatorname{sign}(\mu_i^k) \min\left\{1, \frac{\lambda^k|\mu_i^k|}{\alpha}\right\}.$$

From Eq. (10.12) we see that

$$f^* = \frac{\lambda^k \mu^k - \alpha v^*}{\|\lambda^k \mu^k - \alpha v^*\|_2}.$$

21 Sparse PCA

Input: data matrix $X \in \mathbb{R}^{m \times n}$, sparsity parameter α, accuracy ϵ
Initialization: $f^0 \in \mathbb{R}^n$ with $S(f^0) = 1$, $\lambda^0 = F(f^0)$, $\mu^0 = \frac{\Sigma f^0}{\|Xf^0\|_2}$
repeat
  $g_i^{k+1} = \operatorname{sign}(\mu_i^k)\big(\lambda^k \left|\mu_i^k\right| - \alpha\big)_+$
  $f^{k+1} = \frac{g^{k+1}}{\left\|Xg^{k+1}\right\|_2}$
  $\lambda^{k+1} = (1-\alpha)\left\|f^{k+1}\right\|_2 + \alpha\left\|f^{k+1}\right\|_1$
  $\mu^{k+1} = \frac{\Sigma f^{k+1}}{\left\|Xf^{k+1}\right\|_2}$
until $\frac{\left|\lambda^{k+1}-\lambda^k\right|}{\lambda^k} < \epsilon$

Using that $\lambda^k \mu_i^k - \alpha v_i^* = \operatorname{sign}(\mu_i^k)\ \big(\lambda^k |\mu_i^k| - \alpha\big)_+$, we obtain the result. □

Note that s is just a scaling factor. Thus we can omit it and obtain the simple and efficient scheme to compute sparse loading vectors shown in Alg. 21.

While the derivation is quite different, Journée et al. [2010] propose an algorithm for sparse PCA which turns out to be similar to Alg. 21. In contrast to our method, the algorithm by Journée et al. [2010] works on $\mathbb{R}^m$ instead of $\mathbb{R}^n$. At each step k, variables $z^k, y^k \in \mathbb{R}^m$ are updated as follows:

$$z_i^{k+1} = \sum_{j=1}^{n} X_{ij} \operatorname{sign}(X^T y^k)_j \big(\left|X^T y^k\right|_j - \alpha\big)_+$$
$$y^{k+1} = \frac{z^{k+1}}{\left\|z^{k+1}\right\|_2}.$$

On the other hand, the update step for the variables $g^k, f^k \in \mathbb{R}^n$ in Alg. 21 can be rewritten as (eliminating the variable μ^k),

$$g_i^{k+1} = \operatorname{sign}(X^T X f^k)_i \Big(\frac{\lambda^k}{\|Xf^k\|_2}\left|X^T X f^k\right|_i - \alpha\Big)_+$$
$$f^{k+1} = \frac{g^{k+1}}{\left\|Xg^{k+1}\right\|_2}.$$

Thus, while the two algorithms are quite similar, in our formulation the thresholding parameter of the inner problem depends on the current eigenvalue estimate λ^k whereas it is fixed in Journée et al. [2010]. Empirically, this leads to the fact that we need slightly less iterations to converge.

Up to now we have only considered the case $p = 1$, i.e. we have computed one principal component. In the following we will derive a deflation scheme to obtain multiple principal components. As we will see, in each step we will obtain a ratio problem of the same form as in (10.10).

10.3.2 Deflation scheme

In standard PCA the loading vectors are given by the eigenvectors of the covariance matrix, thus they can be obtained using a singular value decomposition. However, this is not the case in sparse PCA. Here, the usual approach is to iterate between two subproblems: first, one finds the largest eigenvector of the given sample covariance matrix. Then, the influence of that eigenvector is eliminated by *deflation*, see e.g. Mackey [2008].

We now develop a deflation scheme for sparse PCA tailored to the adjusted variance criterion (10.9) discussed in Section 10.2.1. We will derive a greedy scheme where the goal is to directly optimize the contribution to the adjusted variance in each step. Given $l-1$ principal components $(z_i)_{i=1,\dots,l-1}$, where $z_i = Xf_i$, the contribution of the l-th component z_l to the adjusted variance is given by $\|z_l - P_{l-1}(z_l)\|_2^2$, where the $P_{l-1}(z_l)$ is the projection on the space spanned by the previous $l-1$ components, as stated in (10.8). Thus, to compute the lth principal component of the algorithm, we maximize the quantity $\|z_l - P_{l-1}(z_l)\|_2^2$. The goal is to find the loading vector $f \in \mathbb{R}^n$ optimizing

$$\begin{aligned} &\max_{f\in\mathbb{R}^n} \ \|Xf - P_{l-1}(Xf)\|_2^2 \\ &\text{subject to } \ \|f\|_2 = 1. \end{aligned} \tag{10.13}$$

We now show that the problem can be reformulated in such a way that we obtain a problem of the same form as in (10.10).

Lemma 10.6. *For $l = 1, \dots, k$, the problem* (10.13) *can be rewritten as*

$$\min_{f\in\mathbb{R}^n} \ \frac{\|f\|_2}{\|X_l\, f\|_2},$$

where $X_l = \begin{cases} X, & l = 1, \\ \left(I - \frac{\widehat{z}_{l-1}\widehat{z}_{l-1}^T}{\|\widehat{z}_{l-1}\|_2^2}\right) X_{l-1}, & l = 2 \dots k, \end{cases}$ *and* $\widehat{z}_l = X_l\, f_l$, $\forall l = 1 \dots k$.

Proof. The projection $\widehat{z}_l = z_l - P_{l-1}(z_l)$ can be written explicitly as

$$\widehat{z}_l = \left(I - \sum_{i=1}^{l-1} \frac{\widehat{z}_i \widehat{z}_i^T}{\|\widehat{z}_i\|_2^2}\right) z_l,$$

where we used the fact that $\left(\frac{\widehat{z}_i}{\|\widehat{z}_i\|}\right)_{i=1}^{l-1}$ forms an orthonormal basis of the space spanned by the first $l-1$ components. Using again the orthonormality, this can be rewritten as

$$\widehat{z}_l = \prod_{i=1}^{l-1} \left(I - \frac{\widehat{z}_i \widehat{z}_i^T}{\|\widehat{z}_i\|_2^2}\right) z_l = \prod_{i=1}^{l-1} \left(I - \frac{\widehat{z}_i \widehat{z}_i^T}{\|\widehat{z}_i\|_2^2}\right) X f_l.$$

Note that here we can change the order of the terms in the product and thus we can now rewrite this as $\widehat{z}_l = X_l f_l$, where

$$X_l = \begin{cases} X, & l = 1, \\ \left(I - \frac{\widehat{z}_{l-1}\widehat{z}_{l-1}^T}{\|\widehat{z}_{l-1}\|_2^2}\right) X_{l-1}, & l = 2 \dots k. \end{cases}$$

Reformulating the constrained problem as a ratio problem and then turning the maximization problem into a minimization problem yields the result. □

Similarly to the case $l = 1$, in order to enforce sparsity, we replace the L_2-norm by a convex combination of an L_1 norm and L_2 norm in the numerator, with sparsity controlling parameter $\alpha_l \in [0, 1]$. One obtains the functional

$$F(f) = \frac{(1 - \alpha_l) \|f\|_2 + \alpha_l \|f\|_1}{\|X_l f\|_2} , \tag{10.14}$$

and thus one has a problem of the same form as the one in (10.10), which can again be solved using Alg. 21. This suggests the following scheme to compute multiple sparse principal components.

22 Sparse PCA - Deflation Scheme

1: **Input:** data matrix X
2: **Initialization:** $X_1 = X$
3: **for all** $l = 1 : k$ **do**
4: $f_l = \arg\min_{f \in \mathbb{R}^n} \frac{(1-\alpha_l)\|f\|_2 + \alpha_l \|f\|_1}{\|X_l f\|_2}$
5: $\widehat{z}_l = X_l f_l$
6: $X_{l+1} = \left(I - \frac{\widehat{z}_l \widehat{z}_l^T}{\|\widehat{z}_l\|_2^2}\right) X_l$
7: **end for**
8: **return** sparse principal components $f_1, \dots f_l$

In each step, we use bisection in the interval $[0, 1]$ to determine the optimal value of the parameter α_l such that the sparsity constraints $\|f_l\|_0 \leq c$, for $c > 0$, are satisfied.

The above scheme is similar to the generalized deflation method discussed in Mackey [2008], which also reformulates the optimization problem in sparse PCA to explicitly reflect the contribution to the maximized objective in each round. The main difference is that in the scheme proposed by Mackey [2008], at each step the projection is performed orthogonal to the space spanned by the previous loading vectors $(f_1)_{i=1,\dots,l-1}$, thus achieving that $f_i^T f_j = 0$, for all $i \neq j$. In order to reflect the adjusted variance criterion discussed above, in the scheme in Alg. 22 the projection is done with respect to the principal components $z_i = X f_i$, thus explicitly enforcing that $X f_i \perp X f_j$, and therefore $f_i^T \Sigma f_j = 0$, for all $i \neq j$.

10.3.3 Variational renormalization

Moghaddam et al. [2006] introduced the following post-processing technique to improve upon given candidate solutions. Given a vector f with the desired sparsity (i.e. $\|f\|_0 \leq c$ for some $c > 0$) at step l of the algorithm, first one computes the largest eigenvector f' of the matrix $\widehat{\Sigma}_l$ obtained by restricting $\Sigma_l = X_l^T X_l$ to the nonzero pattern of f (i.e. the set $C = \{i \in \{1 \dots n\} \mid f_i \neq 0\}$). Then a sparse loading vector $f^* \in \mathbb{R}^n$ is obtained as

$$f_i^* = \begin{cases} f_i', & i \in C, \\ 0, & \text{else}, \end{cases}$$

where we wlog assumed that $\|f\|_2 = \|f'\|_2 = 1$. In the following lemma we show that the vector f^* satisfies the cardinality constraint while achieving at least the same contribution to the adjusted variance as f.

Lemma 10.7. *The vector f^* satisfies $\|f^*\|_0 \leq c$ and $\|Xf - P_{l-1}(Xf)\|_2^2 \leq \|Xf^* - P_{l-1}(Xf^*)\|_2^2$.*

Proof. Assume wlog that the set of nonzero components of f is given by $\{f_1, \dots, f_c\}$. Denote by $\widehat{f} \in \mathbb{R}^c$ and $\widehat{\Sigma}_l \in \mathbb{R}^{c \times c}$ the restriction of f and Σ_l to the indices $1, \dots, c$, i.e. $\widehat{f}_i = f_i, \forall i = 1, \dots, c$ and $(\widehat{\Sigma}_l)_{i,j} = (\Sigma_l)_{i,j}, \forall i, j = 1, \dots, c$. Then one has

$$\begin{aligned} &\|Xf - P_{l-1}(Xf)\|_2^2 = \|X_l f\|_2^2 = f^T \Sigma_L f = \widehat{f}^T \widehat{\Sigma}_L \widehat{f} \\ \leq &\max_{\|g\|_2=1} g^T \widehat{\Sigma}_l g = (f')^T \widehat{\Sigma}_L f' = (f^*)^T \Sigma_l f^* \\ &= \|X_l f^*\|_2^2 = \|Xf^* - P_{l-1}(Xf^*)\|_2^2 . \end{aligned}$$

Moreover, clearly we have by construction $\|f^*\|_0 = \|f\|_0 \leq c$. □

Thus we apply the above variational renormalization scheme after each step in Alg. 22 to improve a given candidate solution f_l.

10.4 Experimental results

We perform two experiments. In the first experiment, we evaluate the performance of our method for $p = 1$ on a number of gene expression data sets. The second experiment deals with the case $p > 1$. We compare the performance of our method for multiple components to various methods on the well-known pitprops benchmark dataset.

10.4.1 Gene expression data

In our first experiment, we evaluate our IPM for sparse PCA on several gene expression datasets. To see the relevance of sparse PCA for this type of data,

let us make a brief excursion into genetics, see e.g. Parmigiani et al. [2003]. The main function of genes is to control the production of proteins in an organism, which is performed in two steps. The genes are coded in strands of deoxyribonucleic acid (DNA). First, in a process called transcription, a strand of messenger ribonucleic acid (mRNA) is copied from a particular segment of DNA. Then, during translation, mRNA is used to assemble the protein as a chain of amino acids. One is now interested in measuring the amount of transcribed mRNA. This is usually done using so-called DNA microarrays, where the expression levels of thousands of genes can be measured simultaneously. Thus the output of these experiments are large-dimensional datasets, consisting of m observations of the expression levels of n genes, where typically $m \ll n$. In order to analyze the data, one is now interested in performing sparse PCA on the large-dimensional datasets. The goal is to find the major directions in the data, while at the same time the loading vectors should consist of only a few significant genes, making them easy to be interpreted by a human.

We now want to compare the performance of our method on several gene expression datasets with two recent algorithms: the L_1 based single-unit power algorithm of Journée et al. [2010] as well as the EM-based algorithm of Sigg and Buhmann [2008], using code published by the authors. In the following we summarize the properties of the datasets used in our experiments.

Dataset	#genes	#observations
GCM	16063	280
Lung2	18117	39
Prostate1	12600	102

We compute the first sparse loading vector for different number of non-zero components p. For all considered datasets, the three methods achieve very similar performance in terms of the trade-off between explained variance and sparsity of the solution. This is illustrated in Fig. 10.2 where we plot the relative variance (relative to maximal possible variance explained with one single component) versus the number of non-zero components for the three datasets. In fact the results are so similar that for each dataset, the plots of all three methods coincide in one line. In Journée et al. [2010] it also has been observed that the best state-of-the-art algorithms produce the same trade-off curve if one uses the same initialization strategy, and applies the variational renormalization step described in the previous section.

10.4.2 Pitprops data

The pitprops data has become a standard benchmark for methods for sparse PCA [Jolliffe et al., 2003, Zou et al., 2006, Moghaddam et al., 2006, Shen and Huang, 2008, Mackey, 2008, Journée et al., 2010]. Originally introduced

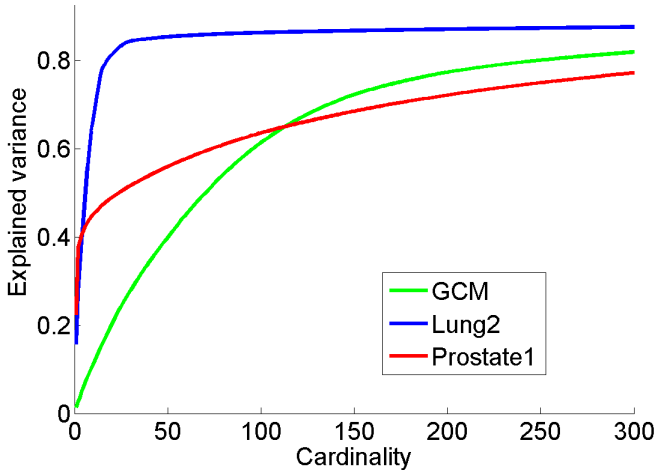

Figure 10.2: Variance (relative to maximal possible variance) versus number of non-zero components for the datasets Lung2, GCM and Prostate1. The same results were obtained for all three investigated methods.

by Jeffers [1967], the dataset contains 180 observations of 13 variables representing some physical properties of mining equipment. Note that for this dataset, only the covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$ is available. Following previous authors (see e.g. Shen and Huang [2008], Journée et al. [2010]), we decompose the covariance matrix into $\Sigma = X^T X$ where $X \in \mathbb{R}^{n\times n}$ is the square root of the matrix Σ. We then apply our method on the "new" data matrix X where the number m of observations has been reduced to 13.

As before we compare our method to the L_1 based single-unit power algorithm (GPower) of Journée et al. [2010] as well as the EM-based algorithm of Sigg and Buhmann [2008], embedded into the same deflation scheme as our method, see Alg. 22. Moreover, we compare to the SPCA method by Zou et al. [2006], as well as thresholding of the loading vectors obtained by standard PCA. In the case of Journée et al. [2010], the optimal parameters where obtained using bisection. For the other methods, we used the default strategies given in the authors' implementations.

In Table 10.1, we show the results for different values of p (number of principal components) as well as different values of c (number of non-zero entries in the loading vectors). We report the cumulative adjusted variance according to (10.9) divided by the total variance (the sum of the eigenvalues of Σ). We observe that our method, GPower and emPCA have very similar performance and achieve the best results in most cases (for small values of c, we see some variation), while PCA and the simple thresholding scheme lead to inferior results in terms of adjusted variance.

Table 10.1: Results on pitprops dataset for different values of p (number of principal components) and c (number of non-zero loadings). We report the cumulative adjusted variance relative to the total variance. Our method, GPower and emPCA have similar performance, while SPCA and thresholding of the standard PCs lead to inferior results in most cases.

$p = 1$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$	$c = 8$
Our method	**0.150**	**0.190**	**0.226**	**0.262**	**0.290**	**0.307**	**0.313**
GPower	**0.150**	0.170	**0.226**	0.246	**0.290**	**0.307**	**0.313**
emPCA	**0.150**	**0.190**	**0.226**	**0.262**	**0.290**	**0.307**	**0.313**
SPCA	0.118	0.188	0.188	0.188	0.186	0.273	0.278
Thresholding	**0.150**	0.177	0.221	0.261	0.289	**0.307**	**0.313**
$p = 2$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$	$c = 8$
Our method	**0.279**	0.329	**0.383**	**0.421**	**0.454**	**0.476**	**0.484**
GPower	**0.279**	0.323	**0.383**	0.407	**0.454**	**0.476**	**0.484**
emPCA	0.256	**0.333**	0.379	**0.421**	**0.454**	**0.476**	**0.484**
SPCA	0.248	0.319	0.320	0.321	0.319	0.415	0.420
Thresholding	0.278	0.318	0.368	0.415	0.450	0.473	0.482
$p = 3$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$	$c = 8$
Our method	**0.393**	**0.471**	**0.535**	**0.565**	**0.602**	**0.622**	**0.630**
GPower	**0.393**	0.430	**0.535**	0.524	**0.602**	**0.622**	**0.630**
emPCA	0.384	**0.471**	**0.535**	**0.565**	**0.602**	**0.622**	**0.630**
SPCA	0.337	0.455	0.460	0.482	0.490	0.568	0.573
Thresholding	0.379	0.409	0.485	0.553	0.589	0.607	0.618
$p = 4$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$	$c = 8$
Our method	0.479	0.555	**0.618**	**0.650**	**0.687**	**0.708**	**0.717**
GPower	**0.482**	0.517	**0.618**	0.611	**0.687**	**0.708**	**0.717**
emPCA	0.466	**0.559**	**0.618**	**0.650**	**0.687**	**0.708**	**0.717**
SPCA	0.426	0.541	0.545	0.567	0.574	0.652	0.657
Thresholding	0.455	0.486	0.564	0.634	0.671	0.692	0.704
$p = 5$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$	$c = 8$
Our method	**0.567**	0.640	**0.689**	**0.721**	**0.754**	0.777	**0.786**
GPower	**0.567**	0.596	**0.689**	0.688	**0.754**	0.775	**0.786**
emPCA	0.551	**0.641**	0.688	**0.721**	0.753	**0.778**	**0.786**
SPCA	0.513	0.629	0.632	0.655	0.667	0.737	0.742
Thresholding	0.509	0.539	0.621	0.696	0.732	0.757	0.770
$p = 6$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$	$c = 8$
Our method	**0.628**	0.706	**0.751**	**0.783**	0.814	0.840	**0.849**
GPower	**0.628**	0.661	**0.751**	0.749	0.814	0.838	**0.849**
emPCA	0.624	**0.707**	**0.751**	**0.783**	**0.815**	**0.841**	**0.849**
SPCA	0.567	0.689	0.701	0.727	0.740	0.806	0.812
Thresholding	0.555	0.579	0.677	0.753	0.793	0.819	0.832

Chapter 11

Conclusions

In this thesis we presented a flexible framework for solving a class of optimization problems known as *constrained fractional set programs (CFSP)*, i.e. the optimization of a ratio of set functions subject to constraints. Optimization problems of this type appear frequently in many areas of machine learning. In this thesis we were mainly interested in applications in clustering and network analysis. The proposed technique involves three steps.

First, the constrained fractional set program is transformed into an equivalent unconstrained fractional set program by incorporating the constraints into the objective via an exact penalty. *Second*, the unconstrained fractional set program is transformed into an equivalent unconstrained continuous optimization problem involving a ratio of non-negative functions. *Third*, a solution of the associated nonlinear eigenproblem is computed using the algorithms introduced in this thesis.

We showed that such a *tight relaxation* of the original constrained fractional set program into an unconstrained continuous optimization problem exists for every constrained minimization of a ratio of non-negative set functions, without any further restrictions on the set functions.

To compute solutions to the corresponding nonlinear eigenproblem and thus optimize the resulting ratios of non-negative functions, we presented a *nonlinear inverse power method* to deal with ratios of p-homogeneous convex functions, as well its generalization *RatioDCA* [Hein and Setzer, 2011], which we further extended to general ratios of differences of p-homogeneous convex functions. While the global optimality of the obtained solution cannot be guaranteed, we proved the convergence to a solution of the associated nonlinear eigenproblem. Moreover, we showed that the loose convex or spectral relaxations are outperformed by a large margin in practice.

Going over to the constrained fractional set programs and the corresponding nonlinear eigenproblems leads to a greater modeling flexibility, as we demonstrated for several applications in data analysis. In particular we developed graph based methods for clustering, local clustering and commu-

nity detection. Our methods based on tight relaxations of the corresponding CFSPs outperform previous methods [Shi and Malik, 2000, Andersen and Lang, 2006, Mahoney et al., 2012] by a large margin. Moreover, the performance of our method was also demonstrated for the sparse PCA problem which was directly modeled in the continuous space. The resulting method was shown to match or outperform state of the art methods [Zou et al., 2006, Sigg and Buhmann, 2008, Journée et al., 2010].

There are still many interesting open questions and directions for further research. While we proved the convergence of our algorithms to a solution of the associated nonlinear eigenproblems, and also observed a strong practical performance in terms of achieved objective values, the convergence to the global optimum could not be guaranteed. It is clear that this cannot be done in general, since in most of the considered examples, the problems are NP hard, and in many cases, for instance the maximum density subgraph problem with upper bound constraint, do not admit a polynomial time approximation scheme [Khot, 2006, Khuller and Saha, 2009]. However, at least for some special problems it should in principle be possible to derive approximation guarantees for our method or modify our method in such a way that approximation guarantees can be given.

There are also still some open questions from a numerical viewpoint. While we showed that the convex inner problem in the nonlinear IPM and RatioDCA does not need to be solved to full accuracy in each step, as long as a point with negative inner objective is found to guarantee descent in the outer objective, it is still unclear what exactly is the optimal strategy for choosing the accuracy to solve the inner problem. Is it better to solve a large sequence of inner problems with low accuracy or a few times with high accuracy? In our empirical observations we observed that typically the best strategy is to use fewer iterations in earlier steps and more iterations in later steps, however a thorough theoretical analysis and thus - if possible - an optimized choice of the parameters is still missing.

One limitation of our methods is that currently we are restricted to the computation of the eigenvectors corresponding to the smallest eigenvalue of the nonlinear eigenproblems (or, by changing the roles of numerator and denominator, the highest eigenvalues). Thus one promising direction with a possibly wide range of applications is to derive methods to compute the higher order eigenvectors of the nonlinear eigenproblems. A particularly interesting case would be the higher nonlinear eigenvectors of the graph p-Laplacian for $p \geq 1$, with possible applications for instance in clustering and dimensionality reduction. For the case $p > 1$, a technique to compute a higher dimensional embedding was proposed by Luo et al. [2010], however the exact connection of the obtained solution to the eigenvectors of the graph p-Laplacian remains unclear.

The question that needs to be answered is what is a suitable min-max principle for nonlinear eigenproblems. Note that while the Ljusternik-

Schnirelmann principle (see e.g. Fučík et al. [1973]) gives a rather abstract way to characterize the higher eigenvalues of nonlinear eigenproblems similar to the Courant-Weinstein min-max principle in the standard linear case, it is currently unclear how this can be used for an explicit construction of the higher eigenvalues and the corresponding eigenvectors. Being able to compute a sequence of higher eigenvectors would no doubt be useful for many applications.

The next question is then how in general the higher order critical points of the nonlinear Rayleigh quotients can be related to the critical points of the original set functions. A first step into this direction was done by Bresson et al. [2012a] for the special case of the Cheeger cut problem. There the authors gave a characterization of the local minima of the functional associated to the Cheeger cut criterion with respect to the local minima of the combinatorial objective. A related question especially relevant for the balanced graph cut applications is whether one can derive tight relaxations of the multi-cut objectives.

Furthermore, there are also many other interesting questions regarding the study of these nonlinear operators. For instance, as pointed out by Alamgir and von Luxburg [2011], while the standard graph Laplacian is related to the commute distance / resistance distance on a graph (see von Luxburg [2007], von Luxburg et al. [2010]), it would be interesting to study the connections between the eigenvectors of the graph p-Laplacian to the more general p-resistances considered by Herbster and Lever [2009] and Alamgir and von Luxburg [2011].

Despite the above open questions, we believe that the generality of our framework already at this point allows for many other applications especially in a graph-based setting. Some work has already been done in that direction. For instance, in addition to the already discussed applications in graph partitioning, local clustering, community detection and team formation, our framework was also used for clustering with pairwise constraints [Rangapuram and Hein, 2012], and learning on hypergraphs [Hein et al., 2013]. Moreover, while the main focus in this thesis was on graph-based applications, set functions appear in many other areas such as feature selection or sensor placement, see e.g. Narasimhan and Bilmes [2004], Krause et al. [2008], Krause and Cevher [2010], Bach [2013]. Thus, there is possibly a much larger class of applications involving ratios of set functions.

Furthermore, in addition to being one step in a technique to solve constrained fractional set programs, the algorithms presented in Chapter 5 for the solution of nonlinear eigenproblems are also useful in their own right, for problems directly modeled in the continuous space. There is a wide range of problems in machine learning which lead to standard linear eigenproblems. Apart from the ones considered in this thesis, examples are given by canonical correlation analysis [Hotelling, 1936], linear discriminant analysis [Fisher, 1936] or modularity optimization [Newman, 2006]. See De Bie et al.

[2005] and references therein for a number of other examples. While the standard variants of these problems, which correspond to linear eigenproblems, can be solved efficiently, the restriction to standard eigenproblems and hence a ratio of quadratic functions is very limiting. In many cases, already a simple modification to the problem formulation with the purpose of adapting it for a given task, such as adjusting the involved functions or adding a regularization term, breaks the structure of the problem, and standard eigensolvers are not applicable anymore.

However, going over to nonlinear eigenproblems, such a fine-tuning to the problem is possible, as shown in this thesis for the sparse PCA problem which was directly modeled in the continuous space. Here, the method was derived as a simple modification of standard PCA where an additional L_1 regularizing term was added to promote sparsity of the solution. Due to the availability of our solvers for nonlinear eigenproblems, this modification was possible and the resulting problem could be solved efficiently. Since a large class of functions can be expressed as a difference of convex functions and thus the RatioDCA is broadly applicable we believe that there is a wide range of other future applications that can be approached by our methods.

Bibliography

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

M. Alamgir and U. von Luxburg. Phase transition in the family of p-resistances. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 379–387, 2011.

N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *J. Comb. Theor., Ser. B*, 38(1):73–88, 1985.

S. Amghibech. Eigenvalues of the discrete p-Laplacian for graphs. *Ars Combin.*, 67:283–302, 2003.

S. Amghibech. Bounds for the largest p-Laplacian eigenvalue for graphs. *Disc. Math.*, 306(21):2762–2771, 2006.

E. D. Andersen and K. D. Andersen. The Mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pages 197–232. Springer, 2000.

R. Andersen. Finding large and small dense subgraphs. *CoRR*, abs/cs/0702032, 2007.

R. Andersen and K. Chellapilla. Finding dense subgraphs with size bounds. In *Proc. Int. Work. Alg. Models Web Graph (WAW)*, pages 25–37, 2009.

R. Andersen and K. Lang. Communities from seed sets. In *Proc. Int. Conf. World Wide Web (WWW)*, pages 223–232, 2006.

R. Andersen and Y. Peres. Finding sparse cuts locally using evolving sets. In *Proc. Ann. ACM Symp. Theor. Comput. (STOC)*, pages 235–244, 2009.

R. Andersen, F. R. K. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Ann. IEEE Symp. Found. Comp. Sci. (FOCS)*, pages 475–486, 2006.

C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.

S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proc. Ann. ACM Symp. Theor. Comput. (STOC)*, pages 222–231, 2004.

K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*. Stanford University Press, 1964.

F. Bach. Learning with submodular functions: A convex optimization perspective. *Found. Trends Mach. Learn.*, 6(2-3):145–373, 2013.

F. Bach and M. I. Jordan. Learning spectral clustering, with application to speech separation. *J. Mach. Learn. Res. (JMLR)*, 7:1963–2001, 2006.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, 2009.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neur. Comput.*, 15:1373–1396, 2002.

M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *Mach. Learn.*, 56(1-3):209–239, 2004.

A. Belloni. Lecture notes for IAP 2005 course "Introduction to Bundle Methods", 2005.

A. Ben-Tal and A. Nemirovski. Non-Euclidean restricted memory level method for large-scale convex optimization. *Math. Program.*, 102(3):407–456, 2005.

Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neur. Comput.*, 16(10):2197–2219, 2004.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

D. P. Bertsekas. Supplementary chapter 6 on convex optimization algorithms. In *Convex Optimization Theory*. Athena Scientific, 2010. URL `http://www.athenasc.com/convexdualitychapter.pdf`.

E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Disc. Appl. Math.*, 123(1-3):155–225, 2002.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

X. Bresson, T. Laurent, D. Uminsky, and J. H. von Brecht. Convergence and energy landscape for Cheeger cut clustering. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 1394–1402, 2012a.

X. Bresson, T. Laurent, D. Uminsky, and J. H. von Brecht. Convergence of a steepest descent algorithm for ratio cut clustering. *CoRR*, abs/1204.6545, 2012b.

X. Bresson, T. Laurent, D. Uminsky, and J. H. von Brecht. Multiclass total variation clustering. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 1421–1429, 2013.

T. Bühler. p-Laplacian based spectral clustering. Master's thesis, Saarland University, Germany, 2009.

T. Bühler and M. Hein. Spectral clustering based on the graph p-Laplacian. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 81–88, 2009a.

T. Bühler and M. Hein. Supplementary material to "Spectral clustering based on the graph p-Laplacian". `http://www.ml.uni-saarland.de/Publications/BueHei09tech.pdf`, 2009b.

T. Bühler, S. S. Rangapuram, S. Setzer, and M. Hein. Constrained fractional set programs and their application in local clustering and community detection. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 624–632, 2013.

P. Buser. Cubic graphs and the first eigenvalue of a Riemann surface. *Math. Zeit.*, 162(1):87–99, 1978.

J. Cadima and I. T. Jolliffe. Loading and correlations in the interpretation of principal components. *J. Appl. Stat.*, 22:203–214, 1995.

A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imag. Vis.*, 20(1-2):89–97, 2004.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.*, 40(1):120–145, 2011.

P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. CAD Integr. Circ. Syst.*, 13(9): 1088–1096, 1994.

K.-C. Chang. Variational methods for non-differentiable functionals and their applications to partial differential equations. *J. Math. Anal. Appl.*, 80:102–129, 1981.

M. Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proc. Int. Work. Approx. Alg. Combin. Optim. (APPROX)*, pages 84–95, 2000.

J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In R. C. Gunning., editor, *Problems in Analysis - A symposium in honor of Salomon Bochner*, pages 195–199. Princeton Univ. Press, 1970.

E. W. Cheney and A. A. Goldstein. Newton's method for convex programming and Tchebycheff approximation. *Numer. Math.*, 1(1):253–268, 1959.

F. R. K. Chung. *Spectral Graph Theory*. AMS, 1997.

F. R. K. Chung, A. Grigor'yan, and S.-T. Yau. Higher eigenvalues and isoperimetric inequalities on Riemannian manifolds and graphs. *Comm. Anal. Geom.*, 8:969–1026, 2000.

F. H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley, 1983.

P. L. Combettes and J.-C. Pesquet. A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE J. Sel. Top. Sign. Proc.*, 1(4):564–574, 2007.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4(4):1168–1200, 2005.

T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

S. I. Daitch, J. A. Kelner, and D. A. Spielman. Fitting a graph to vector data. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 201–208, 2009.

A. Daneshgar, H. Hajiabolhassan, and R. Javadi. On the isoperimetric spectrum of graphs and its approximations. *J. Combin. Theor., Ser. B*, 100(4):390 – 412, 2010.

A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation of sparse PCA using semidefinite programming. *SIAM Review*, 49(3), 2007.

A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *J. Mach. Learn. Res. (JMLR)*, 9:1269–1294, 2008.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1457, 2004.

T. De Bie, N. Cristianini, and R. Rosipal. Eigenproblems in pattern recognition. In *Handbook of Geometric Computing*, pages 129–167. Springer, 2005.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. Amer. Soc. Inform. Sci.*, 41(6):391–407, 1990.

I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, pages 269–274, 2001.

I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: Spectral clustering and normalized cuts. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, pages 551–556, 2004.

G. Di Pillo. Exact penalty methods. In E. Spedicato, editor, *Algorithms for Continuous Optimization*, pages 209–253. Kluwer, 1994.

C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pages 107–114, 2001.

E. A. Dinic. Algorithm for solution of a problem of maximum flow in a network with power estimation. *Soviet Math. Doklady*, 11:1277–1280, 1970.

W. Dinkelbach. On nonlinear fractional programming. *Manag. Sci.*, 13(7): 492–498, 1967.

J. Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Trans. Amer. Math. Soc.*, 284(2):pp. 787–794, 1984.

W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, 17(5):420–425, 1973.

J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82: 421–489, 1956.

Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In *Proc. Int. Conf. World Wide Web (WWW)*, pages 461–470, 2007.

P. Drábek. On the variational eigenvalues which are not of Ljusternik-Schnirelmann type. *Abstr. Appl. Anal.*, 2012.

P. Drábek and J. Milota. *Methods of Nonlinear Analysis: Applications to Differential Equations*. Birkhäuser, 2007.

R. Van Driessche and D. Roose. An improved spectral bisection algorithm and its application to dynamic load balancing. *Parall. Comput.*, 21(1): 29–48, 1995.

J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(3):293–318, 1992.

J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

A. Elmoataz, O. Lezoray, and S. Bougleux. Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Trans. Imag. Proc.*, 17(7):1047–1060, 2008.

E. Esser, X. Zhang, and T. F. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imag. Sci.*, 3(4):1015–1046, 2010.

U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

M. Fiedler. Algebraic connectivity of graphs. *Czech. Math. J.*, 23(98):298–305, 1973.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7(7):179–188, 1936.

L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canad. J. Math.*, 8:399–404, 1956.

S. Fortunato. Community detection in graphs. *Phys. Rep.*, 486(3-5):75 – 174, 2010.

S. Fujishige. *Submodular Functions and Optimization*. Elsevier, 2005.

S. Fučík, J. Nečas, J. Souček, and V. Souček. *Spectral analysis of nonlinear operators*. Springer, 1973.

D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian methods: Applications to the Solution of Boundary-Value Problems*. North-Holland, 1983.

A. Gajewar and A. Das Sarma. Multi-skill collaborative teams based on densest subgraphs. In *SIAM Int. Conf. Data Mining (SDM)*, pages 165–176, 2012.

D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *Proc. Int. Conf. Very Large Data Bases (VLDB)*, pages 721–732, 2005.

A. V. Goldberg. Finding a maximum density subgraph. Technical Report UCB/CSD-84-171, EECS Department, UC Berkeley, 1984.

A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988.

T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Img. Sci.*, 2(2):323–343, apr 2009.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

S. Guattery and G. L. Miller. On the quality of spectral separators. *SIAM J. Matrix Anal. Appl.*, 19:701–719, 1998.

L. Hagen and A. B. Kahng. Fast spectral methods for ratio cut partitioning and clustering. In *Int. Conf. Comput. Aided Design (ICCAD)*, pages 10–13, 1991.

K. M. Hall. An r-dimensional quadratic placement algorithm. *Manag. Sci.*, 17(3):219–229, 1970.

T. Hansen and M. Mahoney. Semi-supervised eigenvectors for locally-biased learning. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 2537–2545, 2012.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

M. Hein and T. Bühler. An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 847–855, 2010.

M. Hein and S. Setzer. Beyond spectral clustering - tight relaxations of balanced graph cuts. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 2366–2374, 2011.

M. Hein, S. Setzer, L. Jost, and S. S. Rangapuram. The total variation on hypergraphs - Learning on hypergraphs revisited. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 2427–2435, 2013.

C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10:673–696, 1997.

B. Hendrickson and R. Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM J. Sci. Comput.*, 16 (2):452–469, 1995.

M. Herbster and G. Lever. Predicting the labelling of a graph via minimum p-seminorm interpolation. In *Proc. Conf. Learn. Theor. (COLT)*, 2009.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms II: Advanced theory and bundle methods*. Springer, 1996.

D. S. Hochbaum. The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Oper. Res.*, 56(4):992–1009, 2008.

I. Holopainen and P. M. Soardi. A strong Liouville theorem for p-harmonic functions on graphs. *Ann. Acad. Sci. Fen.*, 22:205–226, 1997.

S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–561, 2006.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educ. Psych.*, 24, 1933.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

J. H. Hubbard and B. B. Hubbard. *Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach*. Prentice Hall, 1998.

T. Jebara, J. Wang, and S.-F. Chang. Graph construction and B-matching for semi-supervised learning. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 441–448, 2009.

J. Jeffers. Two case studies in the application of principal component analysis. *Appl. Stat.*, 16:225–236, 1967.

T. Joachims. Transductive learning via spectral graph partitioning. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 290–297, 2003.

I. T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

I. T. Jolliffe, N. Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *J. Comput. Graph. Stat.*, 12:531–547, 2003.

L. Jost, S. Setzer, and M. Hein. Nonlinear eigenproblems in data analysis - Balanced graph cuts and the RatioDCA-Prox. *CoRR*, abs/1312.5192v1, 2013.

M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res. (JMLR)*, 11:517–553, 2010.

R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.

M. Kargar and A. An. Discovering top-k teams of experts with/without a leader in social networks. In *Proc. ACM Int. Conf. Informat. Knowl. Manag. (CIKM)*, pages 985–994, 2011.

J. E. Kelley. The cutting-plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8(4):pp. 703–712, 1960.

S. Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM J. Comput.*, 36(4), 2006.

S. Khuller and B. Saha. On finding dense subgraphs. In *Int. Colloq. Autom., Lang. Programm. (ICALP)*, pages 597–608, 2009.

V. King, S. Rao, and R. Tarjan. A faster deterministic maximum flow algorithm. *J. Alg.*, 17(3):447–474, 1994.

K. C. Kiwiel. An aggregate subgradient method for nonsmooth convex minimization. *Math. Program.*, 27(3):320–341, 1983.

K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Program.*, 46(1-3):105–122, 1990.

K. C. Kiwiel. On linear-time algorithms for the continuous quadratic knapsack problem. *J. Opt. Theor. Appl.*, 134(3):549–554, 2007.

A. Krause and V. Cevher. Submodular dictionary selection for sparse representation. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 567–574, 2010.

A. Krause, B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *J. Mach. Learn. Res. (JMLR)*, 9:2761–2801, 2008.

G. Lan. Bundle-level type methods uniformly optimal for smooth and non-smooth convex optimization. *Math. Program.*, pages 1–45, 2013.

T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, pages 467–476, 2009.

T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

C. Lemaréchal. Nonsmooth optimization and descent methods. *Research Report RR-78-4. International Institute of Applied Systems Analysis, Laxenburg, Austria*, 1977.

C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Math. Program.*, 69(1-3):111–147, 1995.

J. Leskovec. Stanford large network dataset collection. URL `http://snap.stanford.edu/data/index.html`.

J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Math.*, 6(1):29–123, 2009.

P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

A. Louis, P. Raghavendra, and S. Vempala. The complexity of approximating vertex expansion. In *Ann. IEEE Symp. Found. Comp. Sci. (FOCS)*, pages 360–369, 2013.

L. Lovász. Submodular functions and convexity. *Math. Program.: The State of the Art*, pages 235–257, 1983.

D. Luo, H. Huang, C. H. Q. Ding, and F. Nie. On the eigenvectors of p-Laplacian. *Mach. Learn.*, 81(1):37–51, 2010.

H. Lütkepohl. *Handbook of Matrices*. Wiley, 1997.

L. Mackey. Deflation methods for sparse PCA. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 1–8, 2008.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, volume 1, pages 281–297, 1967.

M. W. Mahoney, L. Orecchia, and N. K. Vishnoi. A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally. *J. Mach. Learn. Res. (JMLR)*, 13:2339–2365, 2012.

M. Maier, U. von Luxburg, and M. Hein. How the result of graph clustering methods depends on the construction of the graph. *ESAIM: Prob. Stat.*, 17:370–418, 2013.

S. Maji, N. K. Vishnoi, and J. Malik. Biased normalized cuts. In *24th IEEE Conf. Comput. Vis. Patt. Recogn. (CVPR)*, pages 2057–2064, 2011.

B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Rev. Franç. Informat. Rech. Opér.*, 4(3):154–158, 1970.

V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. Technical report, DFG Research Center Matheon, 2005.

M. Meila and J. Shi. A random walks view of spectral segmentation. In *Proc. Int. Work. Art. Intell. Stat. (AISTATS)*, 2001.

B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact and greedy algorithms. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 915–922, 2006.

B. Mohar. The Laplacian spectrum of graphs. *Graph Theor. Combin. Appl.*, 2:871–898, 1991.

J. J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Compt. Rend. Acad. Sci. Paris, Sér. A*, 255:2897–2899, 1962.

D. Mugnolo. Parabolic theory of the discrete p-Laplace operator. *Nonlin. Anal. Theor. Meth. Appl.*, 87(0):33 – 60, 2013.

B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harm. Anal.*, 21(1):113–127, 2006.

M. Narasimhan and J. Bilmes. PAC-learning bounded tree-width graphical models. In *Proc. Conf. Uncert. Art. Intell. (UAI)*, pages 410–417, 2004.

A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Doklady*, 27:372–376, 1983.

Y. Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., 2004.

Y. Nesterov. Gradient methods for minimizing composite objective function. CORE report, Catholic University of Louvain, 2007.

Y. Nesterov. Universal gradient methods for convex optimization problems. *Math. Program.*, pages 1–24, 2014.

M. E. J. Newman. Modularity and community structure in networks. *Proc. Nat. Acad. Sci.*, 103(23):8577–8582, 2006.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 849–856, 2001.

W. Oliveira, C. Sagastizábal, and S. Scheimberg. Inexact bundle methods for two-stage stochastic programming. *SIAM J. Optim.*, 21(2):517–544, 2011.

S. Oveis Gharan and L. Trevisan. Approximating the expansion profile and almost optimal local graph clustering. In *Ann. IEEE Symp. Found. Comp. Sci. (FOCS)*, pages 187–196, 2012.

G. Parmigiani, E. S. Garrett, R. A. Irizarry, and S. L. Zeger, editors. *The Analysis of Gene Expression Data: Methods and Software*. Statistics for Biology and Health. Springer, 2003.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosoph. Mag.*, 2(6):559–572, 1901.

A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, 1990.

S. S. Rangapuram and M. Hein. Constrained 1-spectral clustering. In *Proc. Int. Conf. Art. Intell. Stat. (AISTATS)*, pages 1143–1151, 2012.

S. S. Rangapuram, T. Bühler, and M. Hein. Towards realistic team formation in social networks based on densest subgraphs. In *Proc. Int. Conf. World Wide Web (WWW)*, pages 1077–1088, 2013.

P. Richtárik. Approximate level method for nonsmooth convex minimization. *J. Optim. Theor. Appl.*, 152(2):334–350, 2012.

R. T. Rockafellar. *Convex analysis*. Princeton University Press, 1970.

S. Roweis. EM algorithms for PCA and SPCA. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 626–632, 1998.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

S. Sadie and G. Grove. *The new Grove dictionary of music and musicians*. Macmillan, 2nd edition, 2001.

B. Saha, A. Hoch, S. Khuller, L. Raschid, and X.-N. Zhang. Dense subgraphs with restrictions and applications to gene annotation graphs. In *Ann. Int. Conf. Res. Comput. Molec. Biol. (RECOMB)*, pages 456–472, 2010.

B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender system – a case study. In *ACM WebKDD Work.*, 2000.

S. Schaible. Fractional programming: Applications and algorithms. *Europ. J. Operat. Res.*, 7(2):111–120, 1981.

M. Schmidt, D. Kim, and S. Sra. Projected Newton-type methods in machine learning. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.

S. Setzer. Operator splittings, Bregman methods and frame shrinkage in image processing. *Int. J. Comput. Vision*, 92(3):265–280, 2011.

H. Shen and J. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *J. Multivar. Anal.*, 99(6):1015–1034, 2008.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22(8):888–905, 2000.

C. D. Sigg and J. M. Buhmann. Expectation-maximization for sparse and non-negative PCA. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 960–967, 2008.

H. D. Simon. Partitioning of unstructured problems for parallel processing. *Comput. Syst. Engin.*, 2(2-3):135–148, 1991.

A. J. Soper, C. Walshaw, and M. Cross. A combined evolutionary search and multilevel approach to graph partitioning. *J. Global Optim.*, 29(2): 225–241, 2004.

D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proc. Ann. ACM Symp. Theor. Comput. (STOC)*, pages 81–90, 2004.

D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *CoRR*, abs/cs/0607105, 2006.

D. A. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Lin. Alg. Appl.*, 421(2-3):284–305, 2007.

D. A. Spielman and S.-H. Teng. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011.

D. A. Spielman and S.-H. Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM J. Comput.*, 42(1):1–26, 2013.

B. K. Sriperumbudur, D. A. Torres, and G. R. G. Lanckriet. Sparse eigen methods by D.C. programming. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 831–838, 2007.

A. Szlam and X. Bresson. Total variation and Cheeger cuts. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 1039–1046, 2010.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Royal Stat. Soc., Ser. B*, 58:267–288, 1994.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *J. Royal Stat. Soc., Ser. B*, 61:611–622, 1999.

U. von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17:395–416, 2007.

U. von Luxburg, A. Radl, and M. Hein. Getting lost in space: Large sample analysis of the resistance distance. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 2622–2630, 2010.

J. Šíma and S. E. Schaeffer. On the NP-completeness of some graph cluster measures. In *Proc. Conf. Current Trends Theor. Pract. Comp. Sci. (SOFSEM)*, pages 530–537, 2006.

C. Walshaw. The graph partitioning archive. URL `http://staffweb.cms.gre.ac.uk/~wc06/partition/`.

Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(11):1101–1113, 1993.

F. Yang and Z. Wei. Generalized Euler identity for subdifferentials of homogeneous functions and applications. *J. Math. Anal. Appl.*, 337:516–523, 2008.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Adv. Neur. Inf. Proc. Syst. (NIPS)*, pages 1601–1608, 2004.

H. Zhang, O. Van Kaick, and R. Dyer. Spectral mesh processing. *Comput. Graph. Forum*, 29(6):1865–1894, 2010.

D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *DAGM-Symp.*, pages 361–368, 2005.

M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, pages 08–34, 2008.

X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 912–919, 2003.

Z. A. Zhu, S. Lattanzi, and V. S. Mirrokni. A local algorithm for finding well-connected clusters. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 396–404, 2013.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. Royal Stat. Soc., Ser. B*, 67:301–320, 2005.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comput. Graph. Stat.*, 15:265–286, 2006.