Journal of Machine Learning Research 7 (2006) 1409–1436 Submitted 10/05; Revised 3/06; Published 7/06

Fast SDP Relaxations of Graph Cut Clustering, Transduction,and Other Combinatorial Problems

Tijl De Bie TIJL.DEBIE@GMAIL.COM
Southampton University, ECS, ISIS
SO17 1BJ, United Kingdom
and
K. U. Leuven, OKP Research Group
Tiensestraat 102
3000 Leuven, Belgium

Nello Cristianini NELLO@SUPPORT-VECTOR.NET
University of Bristol
Department of Engineering Mathematics and Department of Computer Science
Queen's Building, University Walk
Bristol, BS8 1TR, United Kingdom

Editors: Kristin P. Bennett and Emilio Parrado-Hernández

Abstract

The rise of convex programming has changed the face of many research fields in recent years, machine learning being one of the ones that benefitted the most. A very recent developement, the relaxation of combinatorial problems to semi-definite programs (SDP), has gained considerable attention over the last decade (Helmberg, 2000; De Bie and Cristianini, 2004a). Although SDP problems can be solved in polynomial time, for many relaxations the exponent in the polynomial complexity bounds is too high for scaling to large problem sizes. This has hampered their uptake as a powerful new tool in machine learning.

In this paper, we present a new and fast SDP relaxation of the normalized graph cut problem, and investigate its usefulness in unsupervised and semi-supervised learning. In particular, this provides a convex algorithm for transduction, as well as approaches to clustering. We further propose a whole cascade of fast relaxations that all hold the middle between older spectral relaxations and the new SDP relaxation, allowing one to trade off computational cost versus relaxation accuracy. Finally, we discuss how the methodology developed in this paper can be applied to other combinatorial problems in machine learning, and we treat the max-cut problem as an example.

Keywords: convex transduction, normalized graph cut, semi-definite programming, semi-supervised learning, relaxation, combinatorial optimization, max-cut

1. Introduction

Let us assume a data sample $\mathcal{S}$ containing n points is given. Between every pair of samples $(\mathbf{x}_i, \mathbf{x}_j)$, an *affinity measure* $\mathbf{A}(i,j) = a(\mathbf{x}_i, \mathbf{x}_j)$ is defined, making up an *affinity matrix* $\mathbf{A}$. We assume the function a is symmetric and positive, however, no positive definiteness of $\mathbf{A}$ will be necessary, probably making the application domain larger than that of kernel based methods as discussed in e.g. Chapelle et al. (2003); De Bie and Cristianini (2004a).

Graph cut clustering Informally speaking, in this paper we are seeking to divide these data points into two coherent sets, denoted by $\mathcal{P}$ and $\mathcal{N}$, such that $\mathcal{P} \bigcup \mathcal{N} = \mathcal{S}$ and $\mathcal{P} \bigcap \mathcal{N} = \emptyset$. In the fully unsupervised-learning scenario, no prior information is given as to which class the points belong to. A number of approaches to bipartitioning sets of data, known as graph cut clustering approaches, make use of an edge-weighted graph, where the nodes in the graph represent the data points and the edges between them are weighted with the affinities between the data points. Bipartitioning the data set then corresponds to cutting the graph in two parts. Intuitively, the fewer high affinity edges are cut, the better the division into two coherent and mutually different parts will be. In Section 1.1 we recall a few graph cut cost functions that have been proposed in literature.

Graph cut transduction Besides this clustering scenario, we also consider the transduction scenario, where part of the class labels is specified. Transduction has received much attention in the past years as a promising middle ground between supervised and unsupervised learning, but major computational obstacles have so far prevented it from becoming a standard piece in the toolbox of practitioners, despite the fact that many natural learning situations directly translate into a transduction problem. In graph cut approaches, the problem of transduction can naturally be approached by restricting the search for a low cost graph cut to graph cuts that do not violate the label information.

Even more generally, one can consider the case where labels are not exactly specified, but where equivalence or inequivalence constraints (Shental et al., 2004) are given instead, specifying equality or non-equality of the labels respectively.

1.1 Cut, Average Cut and Normalized Cut Cost Functions

Several graph cut cost functions have been proposed in literature in the context of clustering, among which the *cut cost*, the *average cut cost (ACut)* and the *normalized cut cost (NCut)* (Shi and Malik, 2000).

The Cut cost is computationally the easiest to handle in a transduction setting (see Blum and Chawla, 2001), however as clearly motivated in Joachims (2003), it often leads to degenerate results with one of both clusters extremely small. This problem could largely be solved by using the ACut or NCut cost functions, of which the ACut cost seems to be more vulnerable to outliers (atypical data points, meaning that they have low affinity to the rest of the sample). However, both optimizing the ACut and NCut costs are NP-complete problems (Shi and Malik, 2000).

To get around this, *spectral* relaxations of the ACut and NCut optimization problems have been proposed in a clustering (Shi and Malik, 2000; Ng et al., 2002; Cristianini et al., 2002) and more recently also in a transduction setting (Kamvar et al., 2003; Joachims, 2003; De Bie et al., 2004). Xing and Jordan (2003) also proposed an interesting SDP relaxation for the NCut optimization problem in a multiclass *clustering* setting, however, the computational cost to solve this relaxation turns out to be too high to cluster data sets of more than about 150 data points, which makes it impractical in real situations.

1.2 Paper Outline

We should emphasize that we are not as much interested in making claims concerning the usefulness of the normalized graph cut for (constrained) clustering problems. A statistical study of the NCut cost function is still lacking, such that claims are necessarily data-dependent, and hence conflicting opinions exist. Instead, we mainly focus on the algorithmic problem involved in the optimization of the normalized graph cut as an interesting object of study on itself, because of its direct applicability to machine learning algorithms design. Furthermore, we will show how the methodologies presented in the context of the NCut optimization problem have a wider applicability, and can be of use to approximately solve other combinatorial problems as well. Our results are structured as follows.

- In Section 2 we recapitulate the well known spectral relaxation of the NCut problem to an eigenvalue problem. Subsequently, a first main result of this paper is presented, which is an efficiently solvable SDP relaxation of the NCut optimization problem. Lastly, this section contains a methodology to construct a cascade of SDP relaxations, all tighter than the spectral relaxation and looser than the SDP relaxation, and with a computational cost in between the cost of both extremes.
- In Section 3 we introduce the so-called *subspace trick*, and show two of its applications. In Section 3.1 we observe how it enables one to efficiently impose equivalence and inequivalence constraints between the labels on the solution of the relaxations. Hence, also transduction problems with the NCut cost can be tackled efficiently. Section 3.2 contains a second application of the subspace trick, consisting of a further speed-up of the relaxations derived in Section 2.
- Lastly, in Section 4 we illustrate how the relaxation cascade and the subspace trick can be applied to speed up relaxations of other combinatorial problems as well, by applying it to the max-cut problem.

We conclude with empirical results for the normalized cut and for the max-cut problems.

2. Relaxations of the Normalized Graph Cut Problem

The NCut cost function for a partitioning of the sample $\mathcal{S}$ into a positive $\mathcal{P}$ and a negative $\mathcal{N}$ set is given by (as originally denoted in Shi and Malik (2000)):

$$\frac{\text{cut}(\mathcal{P},\mathcal{N})}{\text{assoc}(\mathcal{P},\mathcal{S})} + \frac{\text{cut}(\mathcal{N},\mathcal{P})}{\text{assoc}(\mathcal{N},\mathcal{S})} = \left(\frac{1}{\text{assoc}(\mathcal{P},\mathcal{S})} + \frac{1}{\text{assoc}(\mathcal{N},\mathcal{S})}\right) \cdot \text{cut}(\mathcal{P},\mathcal{N}), \tag{1}$$

where $\text{cut}(\mathcal{P},\mathcal{N}) = \text{cut}(\mathcal{N},\mathcal{P}) = \sum_{i:\mathbf{x}_i\in\mathcal{P},j:\mathbf{x}_j\in\mathcal{N}} \mathbf{A}(i,j)$ is the cut between sets $\mathcal{P}$ and $\mathcal{N}$, and $\text{assoc}(\mathcal{P},\mathcal{S}) = \sum_{i:\mathbf{x}_i\in\mathcal{P},j:\mathbf{x}_j\in\mathcal{S}} \mathbf{A}(i,j)$ the association between sets $\mathcal{P}$ and the full sample $\mathcal{S}$. (Note that in fact $\text{cut}(\mathcal{P},\mathcal{N}) = \text{assoc}(\mathcal{P},\mathcal{N})$.) Intuitively, it is clear that the second factor $\text{cut}(\mathcal{P},\mathcal{N})$ defines how well the two clusters separate. The first factor $\left(\frac{1}{\text{assoc}(\mathcal{P},\mathcal{S})} + \frac{1}{\text{assoc}(\mathcal{N},\mathcal{S})}\right)$ measures how well the clusters are balanced. This specific measure of imbalancedness can be seen to improve robustness against atypical data points:[1]

1. This property seems even more important in the relaxations of NCut based methods: the variables then have even more freedom, often making the methods more vulnerable to outliers.

such outliers have a small cut cost with the other data points, making it beneficial to separate them out into a cluster of their own, which would lead to a useless result in our 2-class setting. However, they also have a small association with the rest of the sample $\mathcal{S}$, which on the other hand increases the cost function. In other words, the NCut cost function promotes partitions that are balanced in the sense that both clusters are roughly equally 'coherent', while at the same time 'distant' from each other. It is this feature that makes it preferable over the ACut cost function.[2]

To optimize this cost function, we reformulate it into algebraic terms using the unknown label vector $\mathbf{y} \in \{-1,1\}^n$, the affinity matrix $\mathbf{A}$, the degree vector $\mathbf{d} = \mathbf{A1}$ and associated matrix $\mathbf{D} = \text{diag}(\mathbf{d})$, and shorthand notations $s_+ = \text{assoc}(\mathcal{P}, \mathcal{S})$ and $s_- = \text{assoc}(\mathcal{N}, \mathcal{S})$.

Observe that $\text{cut}(\mathcal{P}, \mathcal{N}) = \frac{(\mathbf{1}+\mathbf{y})'}{2}\mathbf{A}\frac{(\mathbf{1}-\mathbf{y})'}{2} = \frac{1}{4}\left(-\mathbf{y}'\mathbf{A}\mathbf{y} + \mathbf{1}'\mathbf{A}\mathbf{1}\right) = \frac{1}{4}\mathbf{y}'(\mathbf{D}-\mathbf{A})\mathbf{y}$. Furthermore, $s_+ = \text{assoc}(\mathcal{P}, \mathcal{S}) = \frac{1}{2}\mathbf{1}'\mathbf{A}(\mathbf{1}+\mathbf{y}) = \frac{1}{2}\mathbf{d}'(\mathbf{1}+\mathbf{y})$ and $s_- = \frac{1}{2}\mathbf{d}'(\mathbf{1}-\mathbf{y})$. Then we can write the combinatorial optimization problem as:

$$\begin{aligned} \min_{\mathbf{y}, s_+, s_-} \quad & \frac{1}{4}\left(\frac{1}{s_+} + \frac{1}{s_-}\right) \cdot \mathbf{y}'(\mathbf{D}-\mathbf{A})\mathbf{y} \\ \text{s.t.} \quad & \mathbf{y} \in \{-1,1\}^n, \\ & \left\{ \begin{array}{l} s_+ = \frac{1}{2}\mathbf{d}'(\mathbf{1}+\mathbf{y}) \\ s_- = \frac{1}{2}\mathbf{d}'(\mathbf{1}-\mathbf{y}) \end{array} \right. \Leftrightarrow \left\{ \begin{array}{c} \mathbf{d}'\mathbf{y} = s_+ - s_- \\ \mathbf{d}'\mathbf{1} = s_+ + s_- = s, \end{array} \right. \end{aligned}$$

where we introduced the additional symbol s for the constant $s = s_+ + s_- = \mathbf{d}'\mathbf{1} = \mathbf{1}'\mathbf{D}\mathbf{1}$. In this new notation, the optimization problem becomes:

$$\begin{aligned} \min_{\mathbf{y}, s_+, s_-} \quad & \frac{s}{4 s_+ s_-} \cdot \mathbf{y}'(\mathbf{D}-\mathbf{A})\mathbf{y} \\ \text{s.t.} \quad & \mathbf{y} \in \{-1,1\}^n, \\ & \mathbf{d}'\mathbf{y} = s_+ - s_-, \\ & s_+ + s_- = s. \end{aligned} \tag{2}$$

Unfortunately, the resulting optimization problem is known to be NP-complete. Therefore, we approach the problem by relaxing it to more tractable optimization problems. This is the subject of what follows below.

As a guide for the reader, the main notation is summarized in Table 1. We would like to note that we suppress matrix symmetricity constraints where these can be understood from the context.

2.1 A Spectral Relaxation

We now provide a short derivation of the *spectral* relaxation of the NCut optimization problem as first given in Shi and Malik (2000). Let us introduce the variable $\widetilde{\mathbf{y}}$ defined as:

$$\begin{aligned} \widetilde{\mathbf{y}} &= \sqrt{\frac{s}{4 s_+ s_-}}\left(\mathbf{y} - \mathbf{1}\frac{s_+ - s_-}{s}\right) \\ &= \sqrt{\frac{s}{4 s_+ s_-}}\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{d}'}{s}\right)\mathbf{y}, \end{aligned}$$

2. Note however that when a k-nearest neighbor affinity matrix is used, as in Kamvar et al. (2003), every sample has the same affinity with the remainder of the data set, such that the ACut and the NCut costs become equivalent.

Symbol		Definition		Useful identities
$\mathbf{A}$	$=$	affinity matrix $\in \Re_+^{n\times n}$		
$\mathbf{1}$	$=$	$\{1\}^n$		
$\mathbf{I}$	$=$	$\mathrm{diag}(\mathbf{1})$		
$\mathbf{d}$	$=$	$\mathbf{A1}$		
$\mathbf{D}$	$=$	$\mathrm{diag}(\mathbf{d})$		
s_+	$=$	$\mathrm{assoc}(\mathcal{P},\mathcal{S})$	$=$	$\sum_{i:y_i=1} d_i = \mathbf{d}'\frac{\mathbf{1}+\mathbf{y}}{2} = \mathbf{1}'\mathbf{A}\frac{\mathbf{1}+\mathbf{y}}{2}$
s_-	$=$	$\mathrm{assoc}(\mathcal{N},\mathcal{S})$	$=$	$\sum_{i:y_i=-1} d_i = \mathbf{d}'\frac{\mathbf{1}-\mathbf{y}}{2} = \mathbf{1}'\mathbf{A}\frac{\mathbf{1}-\mathbf{y}}{2}$
s	$=$	$\mathrm{assoc}(\mathcal{S},\mathcal{S})$	$=$	$s_+ + s_- = \mathbf{1}'\mathbf{d} = \mathbf{1}'\mathbf{D1} = \mathbf{1}'\mathbf{A1}$
$\mathbf{y}$	$\in$	$\{-1,1\}^n$		
$\widetilde{\mathbf{y}}$	$=$	$\sqrt{\frac{s}{4s_+s_-}}\left(\mathbf{y} - \mathbf{1}\frac{s_+-s_-}{s}\right)$	$=$	$\sqrt{\frac{s}{4s_+s_-}}\left(\mathbf{I} - \frac{\mathbf{1d}'}{s}\right)\mathbf{y}$
p	$=$	$\frac{4s_+s_-}{s^2}$		
q	$=$	$\frac{1}{p}$		
$\mathbf{\Gamma}$	$=$	$\mathbf{y}\mathbf{y}'$		
$\widehat{\mathbf{\Gamma}}$	$=$	$\frac{1}{p}\mathbf{\Gamma}$	$=$	$q\mathbf{\Gamma}$
$\widetilde{\mathbf{\Gamma}}$	$=$	$\frac{s}{4s_+s_-}\left(\mathbf{I} - \frac{\mathbf{1d}'}{s}\right)\cdot\mathbf{\Gamma}\cdot\left(\mathbf{I} - \frac{\mathbf{1d}'}{s}\right)'$	$=$	$\widetilde{\mathbf{y}}\widetilde{\mathbf{y}}'$
$\mathbf{W}$	$\in$	$\Re^{n\times m}$		
$\mathbf{V}$	$=$	eigenvectors of spectral relaxation		
$\widehat{\mathbf{\Gamma}}$	$=$	$\mathbf{V}\mathbf{M}\mathbf{V}'$		
$\mathbf{R}$	$=$	subspace constraint matrix $\in \Re^{n\times d}$		
$\widehat{\mathbf{\Gamma}}$	$=$	$\mathbf{R}\mathbf{M}\mathbf{R}'$		
		(subspace-constrained label matrix)		
$\mathbf{L}$	$=$	label constraint matrix		
$\widehat{\mathbf{\Gamma}}$	$=$	$\mathbf{L}\mathbf{M}\mathbf{L}'$		
		(label-constrained label matrix)		

Table 1: Notation summary. Note that some equalities should be replaced by approximate equalities depending on the context. Throughout the paper, label matrices are understood to be symmetric and any symmetricity constraints are suppressed for conciseness.

and rewrite the optimization problem in terms of this variable by accordingly substituting $\mathbf{y} = \sqrt{\frac{4s_+s_-}{s}}\widetilde{\mathbf{y}} + \mathbf{1}\frac{s_+-s_-}{s}$:

$$\begin{array}{rl} \min_{\widetilde{\mathbf{y}},s_+,s_-} & \widetilde{\mathbf{y}}'(\mathbf{D}-\mathbf{A})\widetilde{\mathbf{y}} \\ \text{s.t.} & \widetilde{\mathbf{y}} \in \left\{-\sqrt{\frac{s_+}{ss_-}}, \sqrt{\frac{s_-}{ss_+}}\right\}^n, \\ & \mathbf{d}'\widetilde{\mathbf{y}} = 0, \\ & s_+ + s_- = s. \end{array} \tag{3}$$

Proposition 1 *The constraints of optimization problem (3) imply that the* $\mathbf{D}$*-weighted 2-norm of* $\widetilde{\mathbf{y}}$ *is constant and equal to* $\widetilde{\mathbf{y}}'\mathbf{D}\widetilde{\mathbf{y}} = 1$.

Proof

$$\begin{aligned} \widetilde{\mathbf{y}}'\mathbf{D}\widetilde{\mathbf{y}} &= \frac{s}{4s_+s_-}\mathbf{y}'\left(\mathbf{I} - \frac{\mathbf{d}\mathbf{1}'}{s}\right)\mathbf{D}\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{d}'}{s}\right)\mathbf{y} \\ &= \frac{s}{4s_+s_-}\mathbf{y}'\left(\mathbf{D} - \frac{\mathbf{d}\mathbf{d}'}{s}\right)\mathbf{y} \\ &= \frac{s}{4s_+s_-}\left(s - \frac{(s_+-s_-)^2}{s}\right) \\ &= 1. \end{aligned}$$

■

Hence we can add the (redundant) constraint $\widetilde{\mathbf{y}}'\mathbf{D}\widetilde{\mathbf{y}} = 1$ to the optimization problem without altering the result. The spectral relaxation is obtained by doing so, and subsequently dropping the combinatorial constraint on $\widetilde{\mathbf{y}}$. The result is:

$$\textbf{Spectral}\left\{\begin{array}{rl} \min_{\widetilde{\mathbf{y}}} & \widetilde{\mathbf{y}}'(\mathbf{D}-\mathbf{A})\widetilde{\mathbf{y}} \\ \text{s.t.} & \widetilde{\mathbf{y}}'\mathbf{D}\widetilde{\mathbf{y}} = 1, \\ & \mathbf{d}'\widetilde{\mathbf{y}} = 0, \end{array}\right. \tag{4}$$

which is solved by taking the (generalized) eigenvector $\widetilde{\mathbf{y}}$ corresponding to the second smallest generalized eigenvalue σ_2 of the generalized eigenvalue problem $(\mathbf{D}-\mathbf{A})\mathbf{v} = \sigma\mathbf{D}\mathbf{v}$. Note that the smallest generalized eigenvalue is $\sigma_1 = 0$, corresponding to the eigenvector $\frac{1}{\sqrt{s}}\mathbf{1}$.

2.2 An SDP Relaxation

We start from formulation (2), and introduce the notation $\mathbf{\Gamma} = \mathbf{y}\mathbf{y}'$. Then, we can write the equivalent optimization problem:

$$\begin{array}{rl} \min_{\mathbf{\Gamma},s_+,s_-} & \frac{s}{4s_+s_-}\langle\mathbf{\Gamma}, \mathbf{D}-\mathbf{A}\rangle \\ \text{s.t.} & \mathbf{\Gamma} = \mathbf{y}\mathbf{y}', \\ & \mathbf{y} \in \{-1,1\}^n, \\ & \langle\mathbf{\Gamma}, \mathbf{d}\mathbf{d}'\rangle = (s_+-s_-)^2 = (s_++s_-)^2 - 4s_+s_-, \\ & s_+ + s_- = s, \quad s_+ > 0, \quad s_- > 0. \end{array} \tag{5}$$

Note that these constraints imply that $(\mathbf{\Gamma}' =)\mathbf{\Gamma} \succeq \mathbf{0}$ and $\text{diag}(\mathbf{\Gamma}) = \mathbf{1}$ (where $\mathbf{A} \succeq \mathbf{B}$ means that $\mathbf{A}-\mathbf{B}$ is positive semi-definite). Hence we can relax the constraint set by adding these

two redundant constraints (we suppress the symmetricity constraint on $\mathbf{\Gamma}$ from the notation for conciseness), and dropping $\mathbf{\Gamma} = \mathbf{y}\mathbf{y}'$ and $\mathbf{y} \in \{-1, 1\}^n$. (While this is a tight relaxation, tighter relaxations are possible at higher computational cost, see Helmberg (2000).) If we further use the notation $p = \frac{4s_+s_-}{s^2}$, we get:

$$\begin{array}{rl} \min_{\mathbf{\Gamma},p} & \frac{1}{p}\langle \mathbf{\Gamma}, \frac{\mathbf{D}-\mathbf{A}}{s}\rangle \\ \text{s.t.} & \mathbf{\Gamma} \succeq \mathbf{0}, \\ & \operatorname{diag}(\mathbf{\Gamma}) = \mathbf{1}, \\ & \frac{1}{s^2}\langle \mathbf{\Gamma}, \mathbf{d}\mathbf{d}'\rangle = 1 - p, \\ & 0 < p \leq 1. \end{array} \tag{6}$$

By once again reparameterizing with $\widehat{\mathbf{\Gamma}} = \frac{\mathbf{\Gamma}}{p}$ and $q = 1/p$, we obtain:

$$\begin{array}{rl} \min_{\widehat{\mathbf{\Gamma}},q} & \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{D}-\mathbf{A}}{s}\rangle \\ \text{s.t.} & \widehat{\mathbf{\Gamma}} \succeq \mathbf{0}, \\ & \operatorname{diag}(\widehat{\mathbf{\Gamma}}) = q\mathbf{1}, \\ & \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{d}\mathbf{d}'}{s^2}\rangle = q - 1, \\ & q \geq 1. \end{array}$$

Note that $\langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{d}\mathbf{d}'}{s^2}\rangle \geq 0$ such that the constraint $\langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{d}\mathbf{d}'}{s^2}\rangle = q - 1$ implies the inequality constraint $q \geq 1$. Hence it does not need to be mentioned explicitly. The result is an optimization problem with a linear objective, $n + 1$ linear equality constraints, and a PSD constraint on a matrix of size n that is linear in the parameters. Hence, we have reshaped the relaxed problem into a standard SDP formulation.

The Lagrange dual We will now derive the dual of this optimization problem, as it will be helpful in the theoretical understanding of the optimization problem as well as for its implementation. To this end we use a symmetric matrix $\mathbf{\Xi} \in \Re^{n\times n}$, a vector $\boldsymbol{\lambda} \in \Re^n$ and a scalar μ as *Lagrange multipliers* (also called *dual variables* in the sequel). Then we can write the Lagrangian as:

$$\begin{aligned} \mathcal{L}(\widehat{\mathbf{\Gamma}}, q, \mathbf{\Xi}, \boldsymbol{\lambda}, \mu) &= \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{D}-\mathbf{A}}{s}\rangle - \langle \widehat{\mathbf{\Gamma}}, \mathbf{\Xi}\rangle - \boldsymbol{\lambda}'\left(\operatorname{diag}(\widehat{\mathbf{\Gamma}}) - q\mathbf{1}\right) - \mu\left((q-1) - \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{d}\mathbf{d}'}{s^2}\rangle\right) \\ &= \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{D}-\mathbf{A}}{s} - \mathbf{\Xi} - \operatorname{diag}(\boldsymbol{\lambda}) + \mu\frac{\mathbf{d}\mathbf{d}'}{s^2}\rangle + q(\mathbf{1}'\boldsymbol{\lambda} - \mu) + \mu, \end{aligned}$$

and the primal optimization problem is equivalent with:

$$\operatorname{opt}_{\text{primal}} = \min_{\widehat{\mathbf{\Gamma}},q}\left[\max_{\mathbf{\Xi}\succeq\mathbf{0},\boldsymbol{\lambda},\mu} \mathcal{L}(\widehat{\mathbf{\Gamma}}, q, \mathbf{\Xi}, \boldsymbol{\lambda}, \mu)\right].$$

Indeed, either the primal constraints are fulfilled and then the inner maximization reduces to the primal objective, or the maximum over the dual constraints is unbounded:

$$\max_{\mathbf{\Xi}\succeq\mathbf{0},\boldsymbol{\lambda},\mu} \mathcal{L}(\widehat{\mathbf{\Gamma}}, q, \mathbf{\Xi}, \boldsymbol{\lambda}, \mu) = \begin{cases} \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{D}-\mathbf{A}}{s}\rangle & \text{if the primal constraints are fulfilled, and} \\ \infty & \text{otherwise.} \end{cases}$$

Thus, in order to minimize this maximum, the primal variables $\widehat{\mathbf{\Gamma}}$ and q will be such that the constraints are met.

The so-called *dual optimization problem* is obtained by interchanging the maximization and minimization in this optimization problem:

$$\text{opt}_{\text{dual}} = \max_{\mathbf{\Xi}\succeq\mathbf{0},\boldsymbol{\lambda},\mu} \left[\min_{\widehat{\mathbf{\Gamma}},q} \mathcal{L}(\widehat{\mathbf{\Gamma}}, q, \mathbf{\Xi}, \boldsymbol{\lambda}, \mu) \right].$$

A very useful relation between the primal and dual optima (on its own already warranting the study of the dual problem) is known as *weak duality*, and says that the dual maximum is a lower bound for the primal minimum (see e.g. Boyd and Vandenberghe (2004)). I.e.:

$$\text{opt}_{\text{primal}} \geq \text{opt}_{\text{dual}}.$$

Let us further focus on the dual optimization problem. The inner minimization can be explicitly solved. Indeed, it is easy to see that it is equal to μ if the following conditions hold:

$$\begin{aligned} \frac{\mathbf{D}-\mathbf{A}}{s} - \mathbf{\Xi} - \text{diag}(\boldsymbol{\lambda}) + \mu\frac{\mathbf{d}\mathbf{d}'}{s^2} &= \mathbf{0} \\ \mathbf{1}'\boldsymbol{\lambda} - \mu &= 0, \end{aligned}$$

and unbounded from below otherwise. Following a similar reasoning as above, this implies that these equalities will hold at the optimum. Hence, we obtain as a dual optimization problem:

$$\begin{array}{ll} \max_{\mathbf{\Xi},\boldsymbol{\lambda},\mu} & \mu, \\ \text{s.t.} & \mathbf{\Xi} \succeq \mathbf{0}, \\ & \mathbf{\Xi} = \frac{\mathbf{D}-\mathbf{A}}{s} - \text{diag}(\boldsymbol{\lambda}) + \mu\frac{\mathbf{d}\mathbf{d}'}{s^2}, \\ & \mathbf{1}'\boldsymbol{\lambda} = \mu. \end{array}$$

The matrix $\mathbf{\Xi}$ is easily eliminated from these constraints, which gives us the final formulation. We state both the primal and the dual:

$$\mathbf{P}^{\mathbf{clust}}_{\mathbf{SDP}} \left\{ \begin{array}{ll} \min_{\widehat{\mathbf{\Gamma}},q} & \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{D}-\mathbf{A}}{s} \rangle \\ \text{s.t.} & \widehat{\mathbf{\Gamma}} \succeq \mathbf{0}, \\ & \text{diag}(\widehat{\mathbf{\Gamma}}) = q\mathbf{1}, \\ & \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{d}\mathbf{d}'}{s^2} \rangle = q - 1. \end{array} \right. \qquad \mathbf{D}^{\mathbf{clust}}_{\mathbf{SDP}} \left\{ \begin{array}{ll} \max_{\boldsymbol{\lambda},\mu} & \mu, \\ \text{s.t.} & \frac{\mathbf{D}-\mathbf{A}}{s} - \text{diag}(\boldsymbol{\lambda}) + \mu\frac{\mathbf{d}\mathbf{d}'}{s^2} \succeq \mathbf{0}, \\ & \mathbf{1}'\boldsymbol{\lambda} = \mu. \end{array} \right.$$

Importantly, this relaxation contains only $n+1$ dual variables. It is thanks to this feature that this relaxation leads to a much more efficient algorithm than the one presented in Xing and Jordan (2003). But we postpone a detailed computational study until Section 2.4.

2.3 A Cascade of Relaxations Tighter Than Spectral and Looser Than SDP

Still, in many cases the SDP relaxation is too complex, while the spectral relaxation is computationally feasible but too loose. Whereas numerous efforts have been made in literature to further tighten SDP relaxations of (other) combinatorial problems by adding in

additional constraints and using so-called lifting techniques (see e.g. Anjos and Wolkowicz (2002)), contributions to further relax the SDP problem without considerably degrading the solution and while gaining on the computational side, have remained limited. Here we present a set of such relaxations, and we will show that they hold the middle between the SDP relaxation and the spectral relaxation, both in terms of computational complexity and in terms of accuracy.

The basic observation to be made is the fact that the constraint $\text{diag}(\widehat{\mathbf{\Gamma}}) = q\mathbf{1}$ implies:

$$\mathbf{W}'\text{diag}(\widehat{\mathbf{\Gamma}}) = q\mathbf{W}'\mathbf{1},$$

for $\mathbf{W} \in \Re^{n\times m}$ (which we choose to be of full column rank, so with $1 \leq m \leq n$). Hence, we can relax the constraint $\text{diag}(\widehat{\mathbf{\Gamma}}) = q\mathbf{1}$ to this weaker constraint. The resulting primal and dual optimization problems are:

$$\mathbf{P}^{\mathbf{clust}}_{\mathbf{m\text{-}SDP}} \begin{cases} \min_{\widehat{\mathbf{\Gamma}},q} & \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{D}-\mathbf{A}}{s} \rangle \\ \text{s.t.} & \widehat{\mathbf{\Gamma}} \succeq \mathbf{0}, \\ & \mathbf{W}'\text{diag}(\widehat{\mathbf{\Gamma}}) = q\mathbf{W}'\mathbf{1}, \\ & \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{d}\mathbf{d}'}{s^2} \rangle = q-1. \end{cases} \qquad \mathbf{D}^{\mathbf{clust}}_{\mathbf{m\text{-}SDP}} \begin{cases} \max_{\boldsymbol{\lambda},\mu} & \mu, \\ \text{s.t.} & \frac{\mathbf{D}-\mathbf{A}}{s} - \text{diag}(\mathbf{W}\boldsymbol{\lambda}) \\ & \quad +\mu\frac{\mathbf{d}\mathbf{d}'}{s^2} \succeq \mathbf{0}, \\ & \mu = \mathbf{1}'\mathbf{W}\boldsymbol{\lambda}. \end{cases}$$

The attractive feature of the relaxation cascade is the fact that the number of dual parameters is only $m+1$, as opposed to $n+1$ for the basic SDP relaxation. Hence, for smaller m, the optimization can be carried out more efficiently.

In general, it is clear that a relaxation is tighter than another if the column space of the matrix $\mathbf{W}$ used in the first one contains the full column space of $\mathbf{W}$ of the second. In particular, for $d = n$ the original SDP relaxation is obtained. At the other extreme, for $m = 1$, let us take $\mathbf{W} = \mathbf{d}$. Then essentially the spectral relaxation is obtained.

Theorem 2 *The SDP relaxation from the cascade with $m = 1$ and $\mathbf{W} = \mathbf{d}$ is (essentially) equivalent to the spectral relaxation.*

Proof Let us write $\widehat{\mathbf{\Gamma}} = \mathbf{V}\mathbf{M}\mathbf{V}'$ with $\mathbf{M} \in \Re^{n\times n}$ a symmetric matrix and with the eigenvectors $\mathbf{v}$ of the spectral relaxation $(\mathbf{D}-\mathbf{A})\mathbf{v} = \sigma\mathbf{D}\mathbf{v}$ as the columns of $\mathbf{V}$, in order of increasing eigenvalue σ, and normalized such that $\mathbf{V}'\mathbf{D}\mathbf{V} = \mathbf{I}$. I.e., the first column of $\mathbf{V}(:,1) = \frac{\mathbf{1}}{\sqrt{s}}$, and the second column $\mathbf{V}(:,2) = \widetilde{\mathbf{y}}$ is the relaxed label vector obtained using the spectral relaxation. Then we have that $\mathbf{V}'(\mathbf{D}-\mathbf{A})\mathbf{V} = \mathbf{\Sigma}$, a diagonal matrix with the generalized eigenvalues in ascending order on the diagonal, i.e. $\mathbf{\Sigma}(1,1) = \sigma_1 = 0$ and $\mathbf{\Sigma}(2,2) = \sigma_2$. Using this reparameterization we can rewrite $\mathbf{P}^{\mathbf{clust}}_{\mathbf{m\text{-}SDP}}$ with $\mathbf{W} = \mathbf{d}$ as:

$$\begin{array}{rl} \min_{\mathbf{M},q} & \langle \widehat{\mathbf{\Gamma}}, \frac{\mathbf{D}-\mathbf{A}}{s} \rangle \equiv \langle \mathbf{M}, \frac{\mathbf{\Sigma}}{s} \rangle \\ \text{s.t.} & \widehat{\mathbf{\Gamma}} \succeq \mathbf{0} \Leftrightarrow \mathbf{M} \succeq \mathbf{0}, \\ & \mathbf{d}'\text{diag}(\mathbf{V}\mathbf{M}\mathbf{V}') \equiv \langle \mathbf{D}, \mathbf{V}\mathbf{M}\mathbf{V}' \rangle \equiv \langle \mathbf{I}, \mathbf{M} \rangle = qs \equiv q\mathbf{d}'\mathbf{1}, \\ & \langle \mathbf{V}\mathbf{M}\mathbf{V}', \frac{\mathbf{d}\mathbf{d}'}{s^2} \rangle \equiv \langle \mathbf{M}, \frac{\mathbf{V}'\mathbf{d}\mathbf{d}'\mathbf{V}}{s^2} \rangle \equiv \frac{1}{s}\mathbf{M}(1,1) = q-1, \end{array}$$

or in summary:

$$\begin{array}{rl} \min_{\mathbf{M},q} & \langle \mathbf{M}, \frac{\mathbf{\Sigma}}{s} \rangle \\ \text{s.t.} & \mathbf{M} \succeq \mathbf{0}, \\ & \langle \mathbf{I}, \mathbf{M} \rangle = qs, \\ & \frac{1}{s}\mathbf{M}(1,1) = q-1, \end{array} \tag{7}$$

where we made use of the fact that $\mathbf{d}'\mathbf{V} = \mathbf{1}'\mathbf{D}\mathbf{V} = \sqrt{s}\mathbf{V}(:,1)'\mathbf{D}\mathbf{V} = \begin{pmatrix} \sqrt{s} & 0 & \cdots & 0 \end{pmatrix}$. We can eliminate q from these constraints, and obtain:

$$\begin{array}{rl} \min_{\mathbf{M}} & \langle \mathbf{M}, \frac{\mathbf{\Sigma}}{s} \rangle \\ \text{s.t.} & \mathbf{M} \succeq \mathbf{0}, \\ & \langle \mathbf{I}, \mathbf{M} \rangle = \mathbf{M}(1,1) + s. \end{array} \tag{8}$$

This optimization problem can be solved by inspection: its optimal solution is given by putting $\mathbf{M}(1,1) = s(q-1)$ for any $q \geq 1$, $\mathbf{M}(2,2) = s$, and $\mathbf{M}(1,2) = \mathbf{M}(2,1) = f$ for any $f \in [-s\sqrt{q-1}, s\sqrt{q-1}]$ (to ensure that $\mathbf{M} \succeq \mathbf{0}$). All other entries of $\mathbf{M}$ should be equal to 0. This means that

$$\begin{aligned} \widehat{\mathbf{\Gamma}} &= s\mathbf{V}(:,2)\mathbf{V}(:,2)' + s(q-1)\mathbf{V}(:,1)\mathbf{V}(:,1)' + f\mathbf{V}(:,2)\mathbf{1}' + f\mathbf{1}\mathbf{V}(:,2)', \\ &= s\widetilde{\mathbf{y}}\widetilde{\mathbf{y}}' + (q-1)\mathbf{1}\mathbf{1}' + f\widetilde{\mathbf{y}}\mathbf{1}' + f\mathbf{1}\widetilde{\mathbf{y}}'. \end{aligned}$$

The value of the optimum is equal to $\mathbf{\Sigma}(2,2) = \sigma_2$, the smallest nonzero generalized eigenvalue. The value of $\widehat{\mathbf{\Gamma}}$ is essentially equivalent to the vector $\widetilde{\mathbf{y}}$ from the spectral relaxation. ■

This result shows that, while the actual choice of how to choose the matrix $\mathbf{W}$ in the relaxation cascade is basically free, for interpretability it is reasonable that $\mathbf{d}$ is within its column space (as only then all relaxations in the cascade are tighter than the spectral relaxation). Well-motivated choices for $\mathbf{W}$ exist, and we will construct one in Section 4.

2.4 Discussion

So far we have introduced a cascade of relaxations of the normalized cut problem, the loosest of which is equivalent to the spectral relaxation. For each SDP relaxation we have derived a dual version, the optimum of which is a lower bound for the primal optimum (weak duality).

In this section we go further into the duality aspects of the SDP problems. In particular, we investigate whether strong duality holds, which would imply that the primal and the dual optima are *equal* to each other. Additionally, this allows us to get a better insight in the relation between the spectral relaxation and the cascade of SDP relaxations.

2.4.1 Strong Duality

Let us investigate whether the dual optimum opt_{dual} is equal to the primal optimum $\text{opt}_{\text{primal}}$, instead of merely a lower bound as guaranteed by the weak duality. If the primal and dual optima are equal to each other, one says that *strong duality* holds. Slater's condition gives a sufficient condition for strong duality to hold.

Lemma 3 (Slater's condition) *Strong duality holds if the primal problem is convex and the primal constraints are strictly feasible. Then the primal and dual optima are equal to each other.*

Hereby, strict feasibility means that a matrix $\widehat{\mathbf{\Gamma}} \succ \mathbf{0}$ along with a value for q satisfying the equality constraints can be found. As SDP problems are convex, the first condition is certainly fulfilled. That the primal constraints in our SDP relaxations are strictly feasible can be seen by construction: choose $\widehat{\mathbf{\Gamma}} = q\mathbf{I} \succ \mathbf{0}$ and $q = 1/(1 - \frac{\mathbf{d}'\mathbf{d}}{s^2})$. Hence:

Proposition 4 *The primal optimization problems* $\mathbf{P}^{\mathbf{clust}}_{\mathbf{SDP}}$ *and* $\mathbf{P}^{\mathbf{clust}}_{\mathbf{m\text{-}SDP}}$ *are strictly feasible. I.e. the Slater condition is fulfilled, and the primal and dual optima are equal to each other:*

$$opt_{dual} = opt_{primal}.$$

If the dual constraints are also strictly feasible, duality theory teaches us that the primal optimum is achieved for a finite value of the variables (see e.g. Helmberg (2000)). However, the following remark answers negatively to this presupposition:

Remark 5 *The dual optimization problems* $\mathbf{D}^{\mathbf{clust}}_{\mathbf{SDP}}$ *and* $\mathbf{D}^{\mathbf{clust}}_{\mathbf{m\text{-}SDP}}$ *are not strictly feasible. Indeed,* $\mathbf{1}'\left(\frac{\mathbf{D}-\mathbf{A}}{s} - diag(\boldsymbol{\lambda}) + \mu\frac{\mathbf{d}\mathbf{d}'}{s^2}\right)\mathbf{1} = 0$ *for all* μ *and* $\boldsymbol{\lambda}$ *satisfying the constraint* $\mu = \mathbf{1}'\boldsymbol{\lambda}$*, and* $\mathbf{1}'\left(\frac{\mathbf{D}-\mathbf{A}}{s} - diag(\mathbf{W}\boldsymbol{\lambda}) + \mu\frac{\mathbf{d}\mathbf{d}'}{s^2}\right)\mathbf{1} = 0$ *for all* μ *and* $\boldsymbol{\lambda}$ *satisfying the constraint* $\mu = \mathbf{1}'\mathbf{W}\boldsymbol{\lambda}$*. This means that the PSD constrained matrix is never strictly positive definite. Correspondingly, the primal optimum is not achieved for a finite value of the variables.*

In particular, note that if $\widehat{\mathbf{\Gamma}}$ is a feasible point of optimization problem $\mathbf{P}^{\mathbf{clust}}_{\mathbf{SDP}}$ or of $\mathbf{P}^{\mathbf{clust}}_{\mathbf{m\text{-}SDP}}$, then also $\widehat{\mathbf{\Gamma}} + x\mathbf{1}\mathbf{1}'$ with $x \geq 0$ is a feasible point with the *same* value of the objective. The consequence is that the optimum will be achieved for matrix $\widehat{\mathbf{\Gamma}}$ with an infinitely large constant component, and hence with q infinitely large. Indeed, increasing q never increases the objective for a fixed value of $\widehat{\mathbf{\Gamma}}$ as the Remark above shows. This also means that the minimum over $\widehat{\mathbf{\Gamma}}$ can only be smaller for q larger, such that the minimum over both q and $\widehat{\mathbf{\Gamma}}$ is obtained for q unboundedly large. What does this mean? A more in-depth study of the relation between the spectral and SDP relaxations makes things clear.

2.4.2 How Much Tighter Are the SDP Relaxations?

We have already shown in Theorem 2 that the SDP relaxation from the cascade with $\mathbf{W} = \mathbf{d}$ is equivalent to the spectral relaxation. Here we prove an even stronger theorem that relates the solution of the basic SDP relaxation, which is the tightest of all, to the spectral one. Our insights gained in the previous section are of help here: the fact that the primal optimum is attained for q approaching infinity will be crucial in the proof. We sketch the proof, which follows a similar reasoning as in the proof of Theorem 2, in Appendix.

Theorem 6 *Also the solution of the basic SDP relaxation* $\mathbf{P}^{\mathbf{clust}}_{\mathbf{SDP}}$ *is essentially equivalent to the spectral relaxation. More specifically, the solution is given by:*

$$\widehat{\mathbf{\Gamma}} \;=\; s\widetilde{\mathbf{y}}\widetilde{\mathbf{y}}' + (q-1)\mathbf{1}\mathbf{1}' + \mathbf{m}\mathbf{1}' + \mathbf{1}\mathbf{m}',$$

with $q \to \infty$*, and* $\mathbf{m}$ *such that* $diag(\widehat{\mathbf{\Gamma}}) = q\mathbf{1}$ *and* $\mathbf{m}'\mathbf{d} = 0$*.*

This result is essentially equivalent with the result from the spectral relaxation, if we ignore the infinitely large constant matrix, and the two rank 2 matrix $\mathbf{m}'\mathbf{1} + \mathbf{1}'\mathbf{m}$ that merely makes the diagonal of the label matrix equal to a constant. A very similar theorem holds for the relaxations from the cascade $\mathbf{P}^{\mathbf{clust}}_{\mathbf{m\text{-}SDP}}$.

So, does this mean that none of the SDP relaxations is tighter than the spectral relaxation? Certainly not: the constraint set is clearly much tighter, as is obvious by looking at the relaxation cascade where constraints on the diagonal of $\widehat{\mathbf{\Gamma}}$ can explicitly be added

or omitted. However, all constraints except for the ones that are also present in the spectral relaxation are inactive. If additional constraints are imposed on the problem, some of the inactive constraints may become active, such that the tightness of the SDP relaxations starts paying off.

2.4.3 Additional Constraints on the SDP problems

A first approach is to introduce an upper bound on q as an additional constraint. Stated in terms of the original variables, this implies that the imbalance $\frac{s^2}{4s_+s_-}$ is upper bounded, which makes sense as this number should be finite for the unrelaxed optimum as well. (If desired, it is easy to compute the maximal value of $\frac{s^2}{4s_+s_-}$ that can be achieved in any bipartitioning of the graph, and this value can be used as a certain upper bound). Interestingly, introducing such an upper bound on q does not affect the correctness of Theorem 2. However, Theorem 6 ceases to hold if q is upper bounded, which was indeed the goal.

Perhaps a more elegant approach is based on the *transductive version* of the NCut relaxation, which we will present in detail in Section 3.1. The transductive version optimizes the same objective while respecting some given labels, and has at most the same number of variables and constraints as in the unconstrained NCut SDP relaxation. However, the dual is automatically strictly feasible as long as at least two data points are labeled differently. We can use this fact as follows. Instead of upper bounding q, one can pick two data points and specify their classes to be different from each other and subsequently solve the transductive NCut SDP relaxation from Section 3.1. It makes sense to pick the two most dissimilar points for this. If needed, the transductive NCut SDP relaxation can be solved for several pairs of data points, up to at most $n-1$ (which, if well chosen, is sufficient to guarantee that at least one of the pairwise inequivalence constraints was correct), although much less than $n-1$ pairs will usually be sufficient to guarantee that with high probability one of the pairwise constraints was correct. Then one proceeds with the solution that achieved the smallest normalized cut value.

2.4.4 Complexity Analysis

We are now ready to study the computational complexity to solve the derived SDP relaxations. The worst-case computational complexity of a pair of primal-dual strictly feasible SDP problems is known to be polynomial, which is achieved by publicly available software tools such as SeDuMi (Sturm, 1999).[3]

In particular for the basic SDP relaxation $\mathbf{P}^{\mathbf{clust}}_{\mathbf{SDP}}$, with (#vars) $= O(n)$ variables (in the dual SDP) and an SDP constraint of size (size SDP) $= O(n)$ the worst case complexity (based on a theoretical analysis of SDP problems without exploiting structure, see Vandenberghe and Boyd (1996)) is given by $O((\#\text{vars})^2(\text{size SDP})^{2.5}) = O(n^{4.5})$, hence the complexity of our basic SDP relaxation with an additional upper bound on q (for dual strict feasibility).

For the SDP cascade $\mathbf{P}^{\mathbf{clust}}_{\mathbf{m\text{-}SDP}}$, it is important to note that the number of dual variables is now only $O(m)$, reducing the worst case complexity down to $O(m^2n^{2.5})$. Hence, m is a

3. Other software tools that are in practice often faster exist, notably SDPLR, which we used for the large-scale experiments (Burer and Monteiro, 2003, 2005).

parameter trading off the tightness of the relaxation with the computational complexity, and can be adapted according to the available computing resources.

2.4.5 Estimating the Label Vector

In the context of the max-cut problem, several techniques have been proposed in literature to construct a good binary label vector based on a label matrix as found by the max-cut SDP relaxation (see e.g. Helmberg (2000) for an overview). Those techniques can be used here as well, and in this paper we use the *randomized rounding* technique.

2.4.6 Bounds on the Unrelaxed Minimum

Let us briefly discuss how the solution of the unrelaxed NCut optimization problem relates to the solution of any of the relaxations. First, since the feasible region is enlarged in the relaxed optimization problem, the relaxed minimum provides a *lower bound* for the minimum of the unrelaxed NCut problem.

On the other hand, the cost of any binary label vector provides an *upper bound* on the minimal cost over all label vectors. Hence, also the label vector as found by the randomized rounding technique will provide such an upper bound. In summary, each of our SDP relaxations allows us to both upper bound and lower bound the minimal NCut cost.

3. The Subspace Trick

In this section we discuss a simple trick that allows one to impose equivalence and inequivalence constraints on the labels in a very natural way. Furthermore, the very same trick leads to a fast approximation of the relaxed NCut optimization problem.

The idea is to reparameterize the label matrix $\widehat{\mathbf{\Gamma}}$ by $\widehat{\mathbf{\Gamma}} = \mathbf{RMR}'$, with $\mathbf{M}$ symmetric and $\mathbf{R}$ a fixed, specified matrix. In this way, we restrict the row and column space of the label matrix $\widehat{\mathbf{\Gamma}}$ to the columns of $\mathbf{R}$.

3.1 Imposing Label Constraints: Transduction and Learning with Side-Information

We first discuss the use of the subspace trick in the transduction scenario, and subsequently extend it to the general semi-supervised learning setting. Here we will use a *label constraint matrix* $\mathbf{L}$ for the matrix $\mathbf{R}$.

The approach of using such label constraint matrices has been used previously by De Bie et al. (2004) to derive a *spectral* relaxation of label-constrained normalized cut cost problems. In the experimental section, we will compare this spectral transduction method with the here derived SDP relaxations.

3.1.1 Transduction

By parameterizing $\widehat{\mathbf{\Gamma}}$ as $\widehat{\mathbf{\Gamma}} = \mathbf{LML}'$, it is straightforward to enforce label constraints in order to achieve a transductive version. Let us assume without loss of generality that the rows and columns of $\mathbf{A}$ are sorted such that the labeled (training) points occur first, with labels given by the label vector $\mathbf{y}_t$, and the unlabeled (test) points thereafter. Then we

define the *label constraint matrix* as:

$$\mathbf{L} = \begin{pmatrix} \mathbf{y}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}.$$

I.e., the first column of the matrix $\mathbf{L}$ consists of the given label vector $\mathbf{y}_t$, and zeros at positions corresponding to the n_{test} test points. The rest of the first block row contains zeros, and the lower right block is an identity matrix of size n_{test}. Then the label constraints can be imposed by observing that any valid $\widehat{\mathbf{\Gamma}}$ must satisfy:

$$\widehat{\mathbf{\Gamma}} = \mathbf{L}\mathbf{M}\mathbf{L}' = \begin{pmatrix} \mathbf{M}(1,1)\mathbf{y}_t\mathbf{y}_t' & \mathbf{y}_t\mathbf{M}(2:n_{\text{test}},1)' \\ \mathbf{M}(2:n_{\text{test}},1)\mathbf{y}_t' & \mathbf{M}(2:n_{\text{test}},2:n_{\text{test}}) \end{pmatrix}.$$

Indeed, rows (columns) corresponding to oppositely labeled training points then automatically are each other's opposite, and rows (columns) corresponding to same-labeled training points are equal to each other.

Using this parameterization we can easily derive the transductive NCut relaxation whose solution will by construction respect the constraints on the training labels:

$$\mathbf{P}^{\mathbf{trans}}_{\mathbf{SDP}} \begin{cases} \min_{\mathbf{M},q} & \langle \mathbf{M}, \mathbf{L}'\frac{\mathbf{D}-\mathbf{A}}{s}\mathbf{L}\rangle \\ \text{s.t.} & \mathbf{M} \succeq \mathbf{0}, \\ & \text{diag}(\mathbf{M}) = q\mathbf{1}, \\ & \langle \mathbf{M}, \mathbf{L}'\frac{\mathbf{d}\mathbf{d}'}{s^2}\mathbf{L}\rangle = q - 1. \end{cases} \qquad \mathbf{D}^{\mathbf{trans}}_{\mathbf{SDP}} \begin{cases} \max_{\boldsymbol{\lambda},\mu} & \mu, \\ \text{s.t.} & s\mathbf{L}'\frac{\mathbf{D}-\mathbf{A}}{s}\mathbf{L} - \text{diag}(\boldsymbol{\lambda}) \\ & \quad +\mu\mathbf{L}'\frac{\mathbf{d}\mathbf{d}'}{s^2}\mathbf{L} \succeq \mathbf{0}, \\ & \mu = \mathbf{1}'\boldsymbol{\lambda}. \end{cases}$$

Note that this is computationally even easier to solve than the unconstrained $\mathbf{P}^{\mathbf{clust}}_{\mathbf{SDP}}$ since the number of dual variables is $n_{\text{test}} + 2$, which decreases with an increasing number of labeled data points.

3.1.2 General Equivalence and Inequivalence Constraints

By using a different label constraint matrix, more general *equivalence and inequivalence constraints* can be imposed (Shental et al., 2004). An equivalence constraint between a pair of data points specifies that they belong to the same class. By extension, one can define an equivalence constraint for a set of points. On the other hand, an inequivalence constraint specifies two data points to belong to opposite classes. It is clear that the transduction scenario is a special case of the scenario where equivalence and inequivalence constraints are given. This large flexibility can be dealt with by using a label constraint matrix of the following form:

$$\mathbf{L} = \begin{pmatrix} \mathbf{1}_{s_1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{1}_{s_2} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{s_3} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{1}_{s_4} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{s_{2p-1}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{1}_{s_{2p}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1}_{s_{2p+1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{1}_{s_{2p+2}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{s_c} \end{pmatrix}.$$

Hereby, the ith row of $\mathbf{L}$ corresponds to the ith data point, in so that samples corresponding to one block row of size s_k are given to belong to the same class by an equivalence constraint (without loss of generality we assume that the samples are organized in this order in the affinity matrix $\mathbf{A}$). Inequivalence constraints are encoded by the first $2p$ block rows: for all $k \leq 2p$, samples from block row k are given to belong to a different class as samples from block row $k+1$. For the last $c-2p$ blocks no inequivalence constraints are given. These blocks will often contain only a single row, meaning that for the corresponding data point no equivalence nor inequivalence constraints are specified.

3.2 Approximating the SDP Relaxation for Speed-Up

Besides for imposing label constraints, the subspace trick can also be used to achieve a further speed-up of the SDP relaxations developed in the previous section. We discuss two different approaches. It is important to stress that both are approximations, and hence no genuine relaxations of the NCut problem anymore.

3.2.1 Using a Coarse Pre-clustering

The semi-supervised learning methodology lends itself to speed up the SDP relaxation itself. A useful approach would be to perform a coarse pre-clustering of the data. The equivalence constraints found by the pre-clustering can then be used as constraints in the constrained SDP relaxation of the NCut problem.

3.2.2 Using the Spectral Relaxation

Assuming that the spectral relaxation performs reasonably well, we know that the optimal label vector will be close to the generalized eigenvector $\widetilde{\mathbf{y}}$ belonging to the smallest nonzero eigenvalue σ_2, plus some constant vector (which is essentially the generalized eigenvector belonging to the smallest eigenvalue $\sigma_1 = 0$). In fact, it is likely that the optimal label vector is close to the space spanned by the eigenvectors corresponding to the d smallest generalized eigenvalues of $(\mathbf{D}-\mathbf{A})\mathbf{v} = \sigma\mathbf{D}\mathbf{v}$. We store these eigenvectors in the columns of the matrix $\mathbf{V}(:, 1:d) \in \Re^{n\times d}$. Then, we can approximate (the optimal value of) $\widehat{\mathbf{\Gamma}}$ by $\widehat{\mathbf{\Gamma}} \approx \mathbf{V}(:,1:d)\mathbf{M}\mathbf{V}(:,1:d)'$.

Since the label vector will only approximately lie in the column space of $\mathbf{V}(:,1:d)$, the equality constraint $\mathbf{W}'\mathrm{diag}(\mathbf{V}(:,1:d)\mathbf{M}\mathbf{V}(:,1:d)') = q\mathbf{W}'\mathbf{1}$ will be infeasible in general. Hence we relax this constraint to an inequality constraint:

$$\mathbf{W}'\mathrm{diag}(\mathbf{V}(:,1:d)\mathbf{M}\mathbf{V}(:,1:d)') \geq q\mathbf{W}'\mathbf{1}.$$

The resulting approximated relaxation then becomes:

$$\mathbf{P}^{\mathbf{appr}}_{\mathbf{SDP}} \left\{ \begin{array}{rl} \min_{\mathbf{M},q} & \langle \mathbf{M}, \frac{\mathbf{\Sigma}(1:d,1:d)}{s} \rangle, \\ \text{s.t.} & \mathbf{M} \succeq \mathbf{0}, \\ & \mathbf{W}'\mathrm{diag}(\mathbf{V}(:,1:d)\mathbf{M}\mathbf{V}(:,1:d)') \geq q\mathbf{W}'\mathbf{1}, \\ & \langle \mathbf{M}, \mathbf{V}(:,1:d)'\mathbf{d}\mathbf{d}'\mathbf{V}(:,1:d) \rangle = q-1. \end{array} \right.$$

$$\mathbf{D}^{\mathbf{appr}}_{\mathbf{SDP}} \left\{ \begin{array}{ll} \max_{\boldsymbol{\lambda},\mu} & \mu, \\ \text{s.t.} & \frac{\boldsymbol{\Sigma}(1:d,1:d)}{s} - \mathbf{V}(:,1:d)'\text{diag}(\mathbf{W}\boldsymbol{\lambda})\mathbf{V}(:,1:d) \\ & \quad +\mu\mathbf{V}(:,1:d)'\frac{\mathbf{d}\mathbf{d}'}{s^2}\mathbf{V}(:,1:d) \succeq \mathbf{0}, \\ & \mu = \mathbf{1}'\mathbf{W}\boldsymbol{\lambda}, \\ & \boldsymbol{\lambda} \geq 0. \end{array} \right.$$

Note that the number of dual variables is equal to $m+1$, and the size of the dual PSD constraint is d. Hence, the computational complexity is now reduced to $O(m^2d^{2.5} + dn^2)$, where the second term arises from the computation of the generalized eigenvectors $\mathbf{V}$ corresponding to the d smallest eigenvalues.

4. Implications Beyond the Normalized Cut

The methodology, developed in this paper in the context of (constrained) NCut bipartitioning, can be used for other combinatorial problems as well. For example, the extension of the developed techniques towards the ACut cost function is straightforward. We briefly discuss the applicability to another example, namely the well-known max-cut problem. For this problem we will also discuss a specific choice of $\mathbf{W}$ in the cascade of SDP relaxations.

4.1 The Max-Cut Problem

The SDP-relaxed max-cut problem is given by (Goemans and Williamson, 1995; Helmberg, 2000):

$$\mathbf{P}^{\mathbf{max\text{-}cut}} \left\{ \begin{array}{ll} \max_{\boldsymbol{\Gamma}} & \frac{1}{4}\langle \boldsymbol{\Gamma}, \mathbf{D}-\mathbf{A}\rangle \\ \text{s.t.} & \boldsymbol{\Gamma} \succeq \mathbf{0}, \\ & \text{diag}(\boldsymbol{\Gamma}) = \mathbf{1}. \end{array} \right. \qquad \mathbf{D}^{\mathbf{max\text{-}cut}} \left\{ \begin{array}{ll} \min_{\boldsymbol{\lambda}} & \mathbf{1}'\boldsymbol{\lambda}, \\ \text{s.t.} & -\frac{1}{4}(\mathbf{D}-\mathbf{A}) + \text{diag}(\boldsymbol{\lambda}) \succeq \mathbf{0}. \end{array} \right.$$

where again $\boldsymbol{\Gamma} \approx \mathbf{y}\mathbf{y}'$ is the label matrix, with $\mathbf{y} \in \{-1,1\}^n$. Just as for the NCut optimization problem, we can relax the constraints on the diagonal to the m constraints $\mathbf{W}'\text{diag}(\boldsymbol{\Gamma}) = \mathbf{W}'\mathbf{1}$ with $\mathbf{W} \in \Re^{n\times m}$. For $\mathbf{W} = \mathbf{1}$ (for $m=1$), the well-known spectral relaxation of max-cut is obtained:

$$\frac{1}{4}(\mathbf{D}-\mathbf{A})\mathbf{v} = \sigma\mathbf{v},$$

where the dominant eigenvector is an approximation for the maximal cut.

Also the subspace trick can readily be applied here, to give rise to label-constrained max-cut relaxations, or to approximations of the max-cut relaxation to control the computational burden. Here, let us define the matrix $\mathbf{V}$ as containing the eigenvectors of the above eigenvalue problem in order of *decreasing* eigenvalue. Then, the approximated max-cut relaxation becomes:

$$\mathbf{P}^{\mathbf{max\text{-}cut\ appr}} \left\{ \begin{array}{ll} \max_{\mathbf{M}} & \frac{1}{4}\langle \boldsymbol{\Gamma}, \mathbf{D}-\mathbf{A}\rangle \\ \text{s.t.} & \mathbf{M} \succeq \mathbf{0}, \\ & \mathbf{W}'\text{diag}(\mathbf{V}(:,1:d)\mathbf{M}\mathbf{V}(:,1:d)') \leq \mathbf{W}'\mathbf{1}. \end{array} \right.$$

$$\mathbf{D}^{\mathbf{max\text{-}cut\ appr}} \left\{ \begin{array}{ll} \min_{\boldsymbol{\lambda}} & \mathbf{1}'\boldsymbol{\lambda}, \\ \text{s.t.} & -\frac{1}{4}(\mathbf{D}-\mathbf{A}) + \text{diag}(\mathbf{W}\boldsymbol{\lambda}) \succeq \mathbf{0}, \\ & \boldsymbol{\lambda} \geq \mathbf{0}. \end{array} \right.$$

A good W for max-cut The matrix $\mathbf{W}$ can essentially be chosen freely, as long as $\mathbf{1}$ is within its column space, in order to maintain the interpretation that for $m = 1$ the spectral relaxation results, and to ensure that for $m > 1$ the relaxation is stricter than for $m = 1$. In particular, we propose to design $\mathbf{W}$ as follows. First, a partition of the data points in m subsets is made. Then, each subset of the partition corresponds to a column of $\mathbf{W}$, and the row-entries in each column that are within the corresponding subset are set equal to 1, the others are kept to 0. The result is that the constraints on the diagonal $\mathbf{W}'\text{diag}(\mathbf{V}(:, 1:d)'\boldsymbol{\lambda}\mathbf{V}(:, 1:d)) \leq \mathbf{W}'\mathbf{1}$ are effectively constraints on sums of subsets of diagonal elements. Clearly, $\mathbf{W1} = \mathbf{1}$ so $\mathbf{1}$ is in $\mathbf{W}$'s column space, as desired. In order to make these constraints as strong as possible, we use the heuristic to put points with a large value in the result of the spectral relaxation in the same subset of the partition. More specifically, we sort the entries of the relaxed label vector from the spectral relaxation, and construct the partition such that the m subsets are (roughly) equally large and such that data points in the same subset occur consecutively in this sorted ordering. This is the approach we use in the empirical results section.

5. Empirical Results

In this section we empirically evaluate the basic SDP relaxation of the NCut problem and its use for transduction. Next, we investigate the cascade of relaxations for the max-cut problem, and the subspace trick to speed up the calculations.

5.1 NCut Clustering and Transduction

In all experiments for NCut clustering and transduction, we use the randomized rounding technique (with 100 random projections) to derive a crisp label vector from the label matrix $\widehat{\mathbf{\Gamma}}$, and K-means on the relaxed label vector $\widetilde{\mathbf{y}}$ obtained from the spectral relaxation. All optimization problems related to the NCut cost function are implemented using the SeDuMi SDP solver (Sturm, 1999).

5.1.1 A Few Toy Problems

The results obtained by using the basic SDP relaxation for a few 2-dimensional clustering problems are summarized in Figure 1. A Gaussian kernel is used with kernel width equal to the average over all data points of the distance to their closest neighbor. In all these cases the resulting label matrix turned out to be indistinguishable from a perfect 1 / -1 label matrix.

5.1.2 Clustering and Transduction on Text

We use the data from De Bie and Cristianini (2004b) to evaluate the clustering and transduction performance of the basic SDP relaxation of the NCut optimization problem. The data set contains 195 articles of the Swiss constitution, each translated in 4 languages (English, French, German and Italian). The articles are grouped into so-called 'Titles', which are topics in the constitution. We use a 20-nearest neighbor affinity matrix $\mathbf{A}$ (meaning that two documents have affinity 1 if they are in the set of 20 nearest neighbors of each other, 0.5 if one is in the set of 20 nearest neighbors of the other but not vice versa, and 0

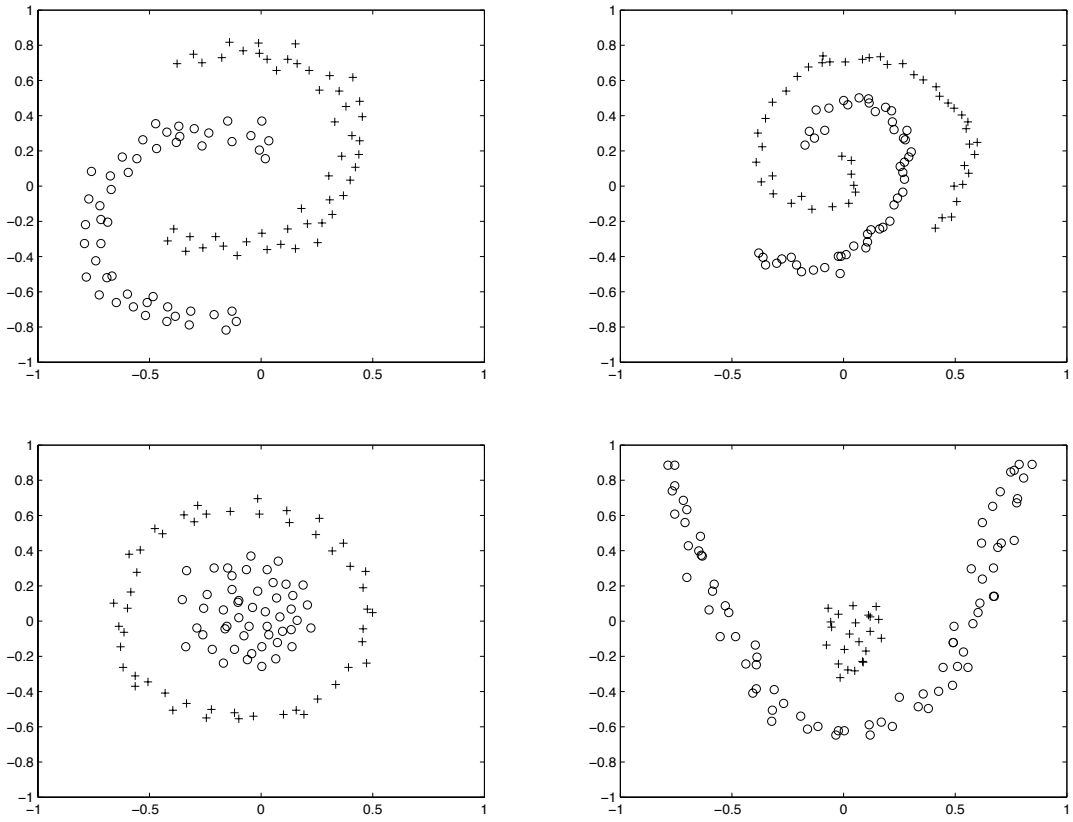

Figure 1: The labeling obtained by the SDP relaxation on 4 toy problems. All results are balanced, except for the last one.

otherwise). The distance used is the cosine distance on the bag of words representation of the documents (computed after stemming and stop word removal), i.e. 1 minus the cosine between both bag of words vectors.

We consider two reasonable divisions of the data as target clusterings. The first division clusters all articles in English with those in French, and those in German with those in Italian. The second clustering is by topic (independent of the language): it clusters those articles in the largest 'Title' together in one cluster, and the articles in all other 'Titles' in the other cluster. Clearly, considering we are using a bag of words kernel, the distinction by language is more natural. However, since there are 4 languages, several bipartitionings are likely to be more or less as natural.

Figures 2 contain the relaxed minimal cost for the transductive spectral relaxation (De Bie et al., 2004) and for the transductive SDP relaxation developed in this paper, as well as the costs corresponding to the label vectors derived from them, as a function of the fraction of data points labeled. The left graph reports the results for the (easy) clustering by language, the right one for the (harder) clustering by topic. Both graphs confirm that the lower bound on the true (unrelaxed) minimum, provided by the SDP relaxation minimum, is consistently (and significantly) tighter than the one provided by the spectral relaxation. Furthermore, the cost of the label vector derived from the spectral relaxation is consistently and significantly larger than the cost of the SDP derived solution. The leftmost points in the figures correspond to the unsupervised case (for the SDP relaxation we used the second approach explained in Section 2.4.3). Note that these unsupervised optima are considerably smaller than the value of the NCut for the true label vector, which is given by the rightmost points in both figures (100% of the data points labeled). This is especially true for the harder problem that ignores languages and clusters based on topic, which is not a surprise. In other words, both target clusterings correspond to a considerably larger cost than the optimal clustering. This result supports the conclusion of Xing and Jordan (2003) that the NCut cost function is not always a good cost function to use for clustering.

On the other hand, even a limited amount of label information seems to guide the prediction to the correct target clustering, even (although to a lesser extent) for the more unnatural clustering by topic. Consider Figure 3, where the test set accuracies for both transduction experiments are shown, again as a function of the fraction of labeled data points. On the left, the performance for the clustering by language is seen to steeply improve for a very small number of labeled data points, to saturate at a level above 0.95. On the right, we see that for the harder less natural division the improvement is less dramatic, and needs more label information, which is to be expected. Interesting to note is that for the easier problem (left figure), the spectral relaxation and the SDP relaxation perform exactly equally, while the SDP problem responds significantly better to label information than the spectral relaxation for the harder problem (right figure). This is evidence for the fact that in a transductive regime, the NCut cost function may be a good one indeed, and it is beneficial to approximate it as well as possible.

5.2 Max-Cut

We use the max-cut problem to conduct an in-depth analysis of the computational consequences of the relaxation cascade and of the subspace trick. We make use of a number of

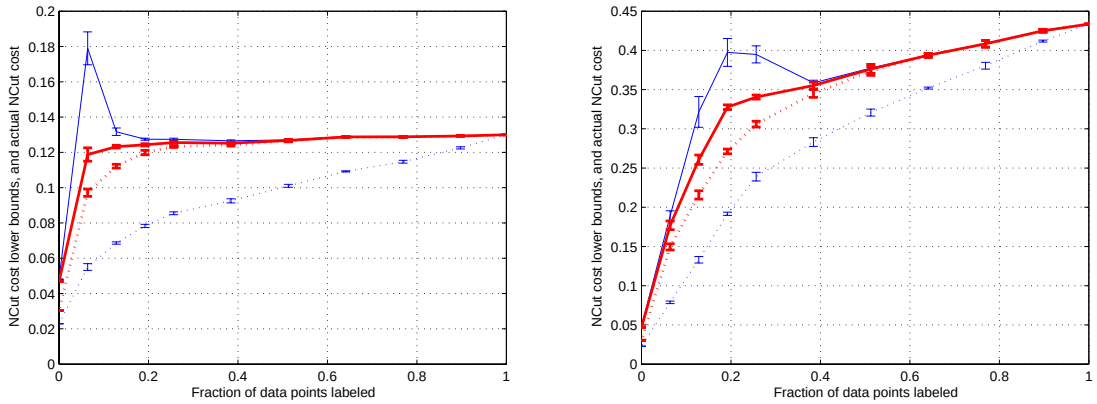

Figure 2: The costs for the best solution over 100 random roundings based on the SDP solution $\widehat{\mathbf{\Gamma}}$ (full bold line), and by performing K-means on the generalized eigenvector $\widetilde{\mathbf{y}}$ of the spectral relaxation (full faint line). This is done in the transductive scenario where the fraction of points labeled is given by the horizontal axis. Hence, the leftmost points in the graph are for the completely unsupervised scenario, and the rightmost points are equal to the cost of the target solution. The dotted lines show the lower bounds provided by the optima of both relaxed problems (bold for the SDP and faint for the spectral relaxation). The plot shows averages (and standard deviations) over 5 random selections of the training set. The left figure is for the clustering by language, the right is for the (harder) clustering by topic.

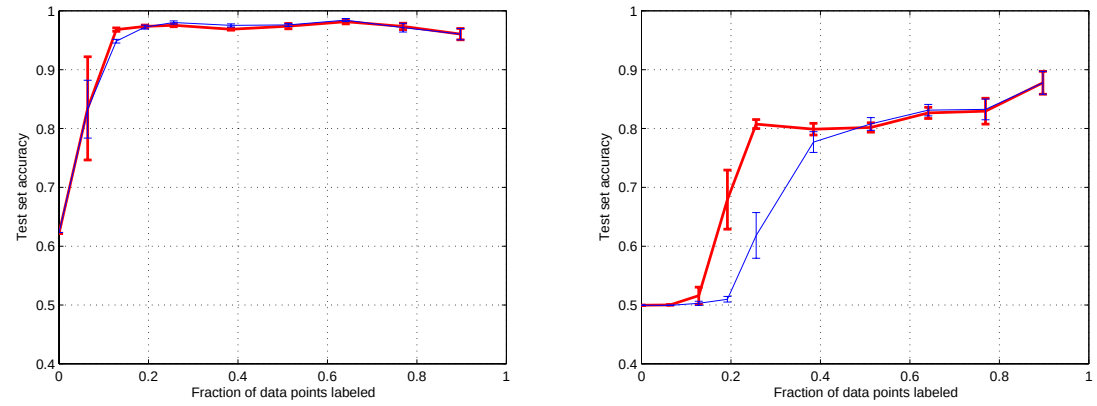

Figure 3: The test set accuracies for the best solution over 100 random roundings based on the SDP solution (bold), and by performing K-means on the generalized eigenvector of the spectral relaxation (faint). Again the horizontal axis represents the fraction of data points labeled. Averages and standard deviations over 5 random selections of the training set are shown. The left figure is for the clustering by language, the right one is for the clustering by topic.

G	$\lvert V\rvert$	$\lvert E\rvert$	density
1	800	19176	6.00
22	2000	19990	1.47
58	5000	29570	0.24
64	7000	41459	0.17
67	10000	20000	0.04
81	20000	40000	0.02

Table 2: The benchmark graphs from the Gset collection. The first column is the identifier of the graph, the second the number of vertices in the graph, the third the number of edges, and the last column shows the edge-density of the graph.

publicly available benchmark data sets from the so-called Gset collection (Helmberg and Rendl, 2000). For a summary of the graphs we used, see Table 2. For these graphs, Figure 4, shows the results of a computational analysis of the relaxation cascade and of the subspace trick as outlined in Section 4.1. In all experiments, a crisp label vector is derived from a relaxed vector (obtained using the spectral relaxation) by simple thresholding around 0, and from a relaxed label matrix by using the randomized rounding technique explained in Section 2.4.5 with 100 random projections (for the SDP relaxations). Motivated by the large size of some of the graphs in the Gset collection, we use the highly effective SDP solver SDPLR (Burer and Monteiro, 2003, 2005) called from within MATLAB in all max-cut experiments on which we report here.

For the relaxation cascade, there is only one parameter to study the effect of: the number of constraints m on the trace of the label matrix. We varied this parameter over all values $1, 2, 4, 8, 16, 32, 64, 128, 256$ and n, where for $m = 1$, the algorithm reduces to the spectral relaxation, and for $m = n$ the well-known SDP relaxation is obtained. In Figures 4, the value of the cut for each of these values of m is plotted as a function of the computation time (full line with cross markers). Average times and cuts over 10 simulations are shown, to account for the randomness of the rounding procedure and effect on the running time of the random initialization of the optimization procedure. Apparently, already a relatively small value for m and correspondingly small increase in computation time results in a significant increase of the cut found. Still, for the two largest graphs in the benchmark, our Pentium 2GHz with 1Mb RAM was unable to solve any of the SDP formulations, for memory reasons, and only one cross is plotted for $m = 1$, the spectral relaxation.

The small dots in the figures give an idea of the effect of the subspace trick, for subspace dimensionality d equal to $d = 2, 4, 8, 16, 32$, in combination with the values for m (except for 1 and n) used as above in the relaxation cascade. I.e., there are $5 \times 8 = 40$ dots in each plot. Clearly the subspace trick allows one to achieve a generally higher cut value at a significantly reduced computational cost. Using the subspace approximation, it is also possible to find a better cut than the one found using the spectral relaxation for the two most challenging problems below in the figure.

Even though the cascade of relaxations empirically appears less efficient in obtaining good approximations to the relaxed optimum, a major disadvantage of the subspace trick is

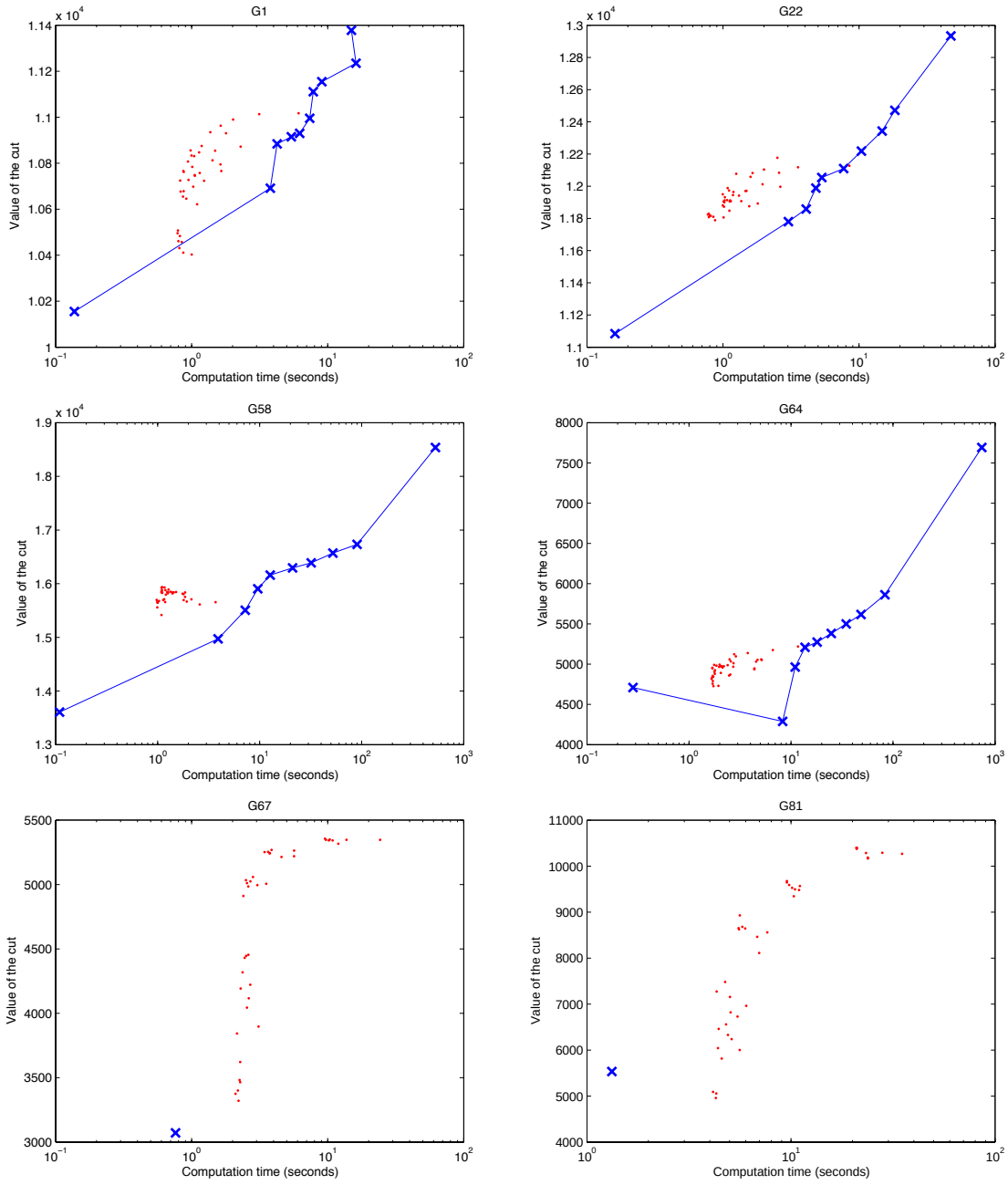

Figure 4: These plots show the value of the cut as a function of the running time for various parameter settings, each figure for another benchmark graph from the Gset collection: G1, G22, G58, G64, G67 and G81. Note the logarithmic scale on the time axis. The crosses correspond to the relaxation cascade, with $m = 1, 2, 4, \ldots, 256, n$. The small dots correspond to the use of the subspace trick for the same values of m and for various dimensionality d of the subspace: $d = 2, 4, 8, 16, 32$. For the last two graphs G67 abd G81, the relaxation cascade requires too much memory to solve on a Pentium 2GHz with 1Gb Ram and is therefore omitted (except for $d = 1$, the spectral relaxation).

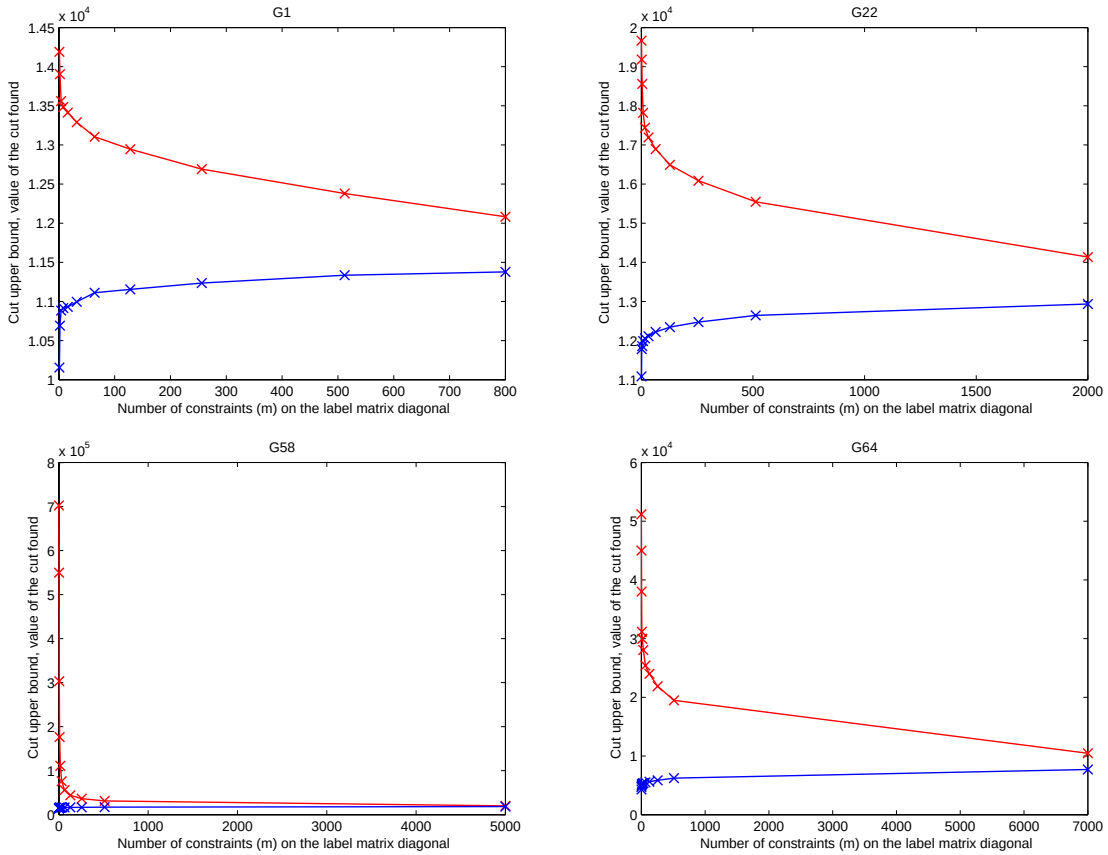

Figure 5: The upper bound on the cut as provided by the SDP relaxations (upper curves), along with the actual cut found (lower curves), for the benchmark graphs G1, G22, G58 and G64.

that it is an approximation and not a genuine relaxation. The result is that the application of the subspace trick makes the relaxed optimum useless to bound the value of the true optimum.[4] So let us now investigate to what extent the cascade of relaxations is helpful in obtaining a bound on the unrelaxed optimum in those cases where computing the full SDP relaxation is too time consuming. Figure 5 contains the value of the max-cut relaxations as a function of the number of constraints m on the diagonal of the label matrix, as well as the actual value of the cut found. One can see that the SDP upper bound on the maximum indeed decreases (that is, tightens) for increasing m. At the same time, for larger m the objective value (the cut cost) for the found label vector increases. Interestingly, it increases (and the upper bound decreases) rather steeply in the beginning and then flattens off, suggesting that a relatively low value for m may often be sufficient in practical cases.

4. Such bounds are of use for example when a Branch&Bound method is employed to find the exact optimum of the combinatorial problem.

6. Conclusions

We proposed a new cascade of SDP relaxations of the NP-complete normalized graph cut optimization problem. On both extremes of the cascade are the well-know spectral relaxation and a newly proposed SDP relaxation. The proposed relaxations directly translate into efficient machine learning algorithms for unsupervised and semi-supervised learning.

The availability of a series of relaxations with different computational complexity and tightness allows one to trade off the computational cost versus accuracy. Furthermore, we introduced the 'subspace trick', which is a simple technique that makes it possible to efficiently impose label constraints on the result of the relaxed optimization problem. Besides this, the subspace trick provides ways to obtain approximate formulations of the relaxed optimization problems with a further reduced computational cost. We believe that an interesting aspect of the paper is the fact that the techniques presented may prove useful in relaxations of other combinatorial problems as well, as witnessed by their application to the max-cut problem.

The application of these efficient approximations to machine learning algorithms might have the potential to finally fulfill the promise of SDP as a powerful new addition to the machine-learning toolbox.

We reported encouraging empirical results for the use of the NCut cost function and more in particular of its newly proposed SDP relaxation for clustering and for semi-supervised learning. Furthermore, we illustrated the use of the cascade of relaxations and of the subspace trick on the max-cut problem.

An interesting research direction opened in this paper is the question which are good and and efficiently computable choices for the matrix $\mathbf{W}$, both for the relaxation cascade and for the subspace approximation that is based on it. An answer to this question may have broad implications in the field of combinatorial optimization and relaxation theory.

An alternative avenue that can be followed to increase the scalability of SDP relaxations can be found in Lang (2004). It is based on the exploiting the ideas behind the SDPLR method (Burer and Monteiro, 2003), and works by explicitly restricting the rank of the label matrix. Further research should clarify potential relations and synergies between their method and the approaches developed in this paper.

Acknowledgments

We are grateful to Sandor Szedmak for some very useful suggestions, and to the anonymous reviewers who considerably improved the quality of this work. Furthermore, we kindly acknowledge the help of Samuel Burer in using the SDPLR method. Nello Cristianini is supported by the NIH grant No. R33HG003070-01. Tijl De Bie acknowledges support from the CoE EF/05/007 SymBioSys, and from GOA/2005/04 (both from the Research Council K.U.Leuven), and from the IST Programme of the European Community under the PASCAL Network of Excellence (IST-2002-506778).

Appendix A. Proof of Theorem 6

While this theorem would be easy to prove by plugging in the result provided in the theorem statement, for the sake of clarity we give here a prove that derives the result rather than posing it.

Proof Let us use the same notation as in the proof of Theorem 2. Then we can rewrite $\mathbf{P}^{\mathbf{clust}}_{\mathbf{SDP}}$ (in the same line as for Theorem 2):

$$\begin{array}{rl} \min_{\mathbf{M},q} & \langle \mathbf{M}, \frac{\mathbf{\Sigma}}{s} \rangle \\ \text{s.t.} & \mathbf{M} \succeq \mathbf{0}, \\ & \text{diag}(\mathbf{V}\mathbf{M}\mathbf{V}') = q\mathbf{1}, \\ & \frac{1}{s}\mathbf{M}(1,1) = q - 1. \end{array}$$

We can eliminate q from these constraints by substituting $q = 1 + \frac{1}{s}\mathbf{M}(1,1)$, and obtain:

$$\begin{array}{rl} \min_{\mathbf{M}} & \langle \mathbf{M}, \frac{\mathbf{\Sigma}}{s} \rangle \\ \text{s.t.} & \mathbf{M} \succeq \mathbf{0}, \\ & \text{diag}(\mathbf{V}\mathbf{M}\mathbf{V}') = \mathbf{1}\left(1 + \frac{1}{s}\mathbf{M}(1,1)\right). \end{array}$$

Let us decompose the left hand side of the last constraint in the following way:

$$\begin{array}{rl} & \text{diag}(\mathbf{V}(:,1)\mathbf{M}(1,1)\mathbf{V}(:,1)') + 2\text{diag}(\mathbf{V}(:,1)\mathbf{M}(1,2:n)\mathbf{V}(:,2:n)') \\ & +\text{diag}(\mathbf{V}(:,2:n)\mathbf{M}(2:n,2:n)\mathbf{V}(:,2:n)') \\ = & \mathbf{1}\frac{1}{s}\mathbf{M}(1,1) + 2\text{diag}(\mathbf{V}(:,1)\mathbf{M}(1,2:n)\mathbf{V}(:,2:n)') \\ & +\text{diag}(\mathbf{V}(:,2:n)\mathbf{M}(2:n,2:n)\mathbf{V}(:,2:n)'). \end{array}$$

We also rewrite the PSD constraint by using the *Schur complement lemma* as:

$$\mathbf{M} \succeq \mathbf{0} \quad \Leftrightarrow \quad \mathbf{M}(2:n,2:n) \succeq \frac{\mathbf{M}(1,:2:n)\mathbf{M}(1,2:n)'}{\mathbf{M}(1,1)}.$$

Besides, as $\mathbf{\Sigma}(1,1) = 0$ and $\mathbf{\Sigma}$ is diagonal, we can write

$$\langle \mathbf{M}, \frac{\mathbf{\Sigma}}{s} \rangle = \langle \mathbf{M}(2:n,2:n), \frac{\mathbf{\Sigma}(2:n,2:n)}{s} \rangle.$$

Then the optimization problem $\mathbf{P}^{\mathbf{clust}}_{\mathbf{SDP}}$ becomes:

$$\begin{array}{rl} \min_{\mathbf{M}} & \langle \mathbf{M}(2:n,2:n), \frac{\mathbf{\Sigma}(2:n,2:n)}{s} \rangle \\ \text{s.t.} & \mathbf{M}(2:n,2:n) \succeq \frac{\mathbf{M}(1,:2:n)\mathbf{M}(1,2:n)'}{\mathbf{M}(1,1)}, \\ & 2\text{diag}(\mathbf{V}(:,1)\mathbf{M}(1,2:n)\mathbf{V}(:,2:n)') + \text{diag}(\mathbf{V}(:,2:n)\mathbf{M}(2:n,2:n)\mathbf{V}(:,2:n)') = \mathbf{1}. \end{array}$$

We can see that the variable $\mathbf{M}(1,1)$ only occurs in the first constraint, and that this constraint becomes less stringent for $\mathbf{M}(1,1) \to \infty$ (note that this corresponds to $q \to \infty$, which should not be a surprise), such that the minimum will be achieved for $\mathbf{M}(1,1)$ unboundedly large. So let us already take the limit for $\mathbf{M}(1,1)$ to infinity, which gives:

$$\begin{array}{rl} \min_{\mathbf{M}} & \langle \mathbf{M}(2:n,2:n), \frac{\mathbf{\Sigma}(2:n,2:n)}{s} \rangle \\ \text{s.t.} & \mathbf{M}(2:n,2:n) \succeq \mathbf{0}, \\ & 2\text{diag}(\mathbf{V}(:,2:n)\mathbf{M}(2:n,1)\mathbf{V}(:,1)') + \text{diag}(\mathbf{V}(:,2:n)\mathbf{M}(2:n,2:n)\mathbf{V}(:,2:n)') = \mathbf{1}. \end{array}$$

The dual of this optimization problem, using Lagrange multipliers $\boldsymbol{\lambda}$ for the equality constraint on the diagonal and the symmetric matrix $\boldsymbol{\Xi}$ for the PSD constraint, is given by:

$$\begin{array}{rl} \max_{\boldsymbol{\lambda},\boldsymbol{\Xi}} & \mathbf{1}'\boldsymbol{\lambda} \\ \text{s.t.} & \boldsymbol{\Xi} = \boldsymbol{\Sigma}(2:n,2:n) - \mathbf{V}(:,2:n)'\text{diag}(\boldsymbol{\lambda})\mathbf{V}(:,2:n), \\ & \boldsymbol{\Xi} \succeq \mathbf{0}, \\ & \mathbf{V}(:,2:n)'\boldsymbol{\lambda} = 0, \end{array}$$

or equivalently:

$$\begin{array}{rl} \max_{\boldsymbol{\lambda}} & \mathbf{1}'\boldsymbol{\lambda} \\ \text{s.t.} & \boldsymbol{\Sigma}(2:n,2:n) - \mathbf{V}(:,2:n)'\text{diag}(\boldsymbol{\lambda})\mathbf{V}(:,2:n) \succeq \mathbf{0}, \\ & \mathbf{V}(:,2:n)'\boldsymbol{\lambda} = 0. \end{array}$$

From the equality constraint we can immediately see that $\boldsymbol{\lambda} \propto \mathbf{d}$, such that $\mathbf{V}(:,2:n)'\text{diag}(\boldsymbol{\lambda})\mathbf{V}(:,2:n) \propto \mathbf{V}(:,2:n)'\mathbf{D}\mathbf{V}(:,2:n) = \mathbf{I}$. Therefrom we can see that $\boldsymbol{\lambda} = \sigma_2\mathbf{d}$ and hence $\boldsymbol{\Xi} = \boldsymbol{\Sigma}(2:n,2:n) - \sigma_2\mathbf{I}$ at the optimum (where σ_2 is used to denote $\boldsymbol{\Sigma}(2,2)$, the smallest generalized eigenvalue of the spectral relaxation that is different from 0). The optimum itself is equal to $\mathbf{1}'\boldsymbol{\lambda} = \sigma_2$.

To determine the value of the primal variables at the optimum, let us now have a look at the Karush Kuhn Tucker (KKT) condition corresponding to the PSD constraint:

$$\langle \mathbf{M}(2:n,2:n), \boldsymbol{\Xi} \rangle = \langle \mathbf{M}(2:n,2:n), \boldsymbol{\Sigma}(2:n,2:n) - \sigma_2\mathbf{I} \rangle \quad = \quad 0$$

The implies that $\mathbf{M}(i,j) = 0$ for all $i \geq 2$ and $j \geq 2$ except for $i = j = 2$. The exact value of $\mathbf{M}(2,2)$ can be derived from the second KKT condition:

$$\begin{array}{rl} & \left(2\text{diag}(\mathbf{V}(:,2:n)\mathbf{M}(2:n,1)\mathbf{V}(:,1)') + \text{diag}(\mathbf{V}(:,2:n)\mathbf{M}(2:n,2:n)\mathbf{V}(:,2:n)') - \mathbf{1}\right)'\boldsymbol{\lambda} \\ = & \left(\frac{2}{\sqrt{s}}\mathbf{V}(:,2:n)\mathbf{M}(2:n,1) + \text{diag}(\mathbf{V}(:,2:n)\mathbf{M}(2:n,2:n)\mathbf{V}(:,2:n)') - \mathbf{1}\right)'\sigma_2\mathbf{d} \\ = & 0 \end{array}$$

From this follows that $\langle \mathbf{D}, \mathbf{V}(:,2:n)\mathbf{M}(2:n,2:n)\mathbf{V}(:,2:n)' \rangle = \langle \mathbf{I}, \mathbf{M}(2:n,2:n) \rangle = s$. Thus, we obtain that $\mathbf{M}(2,2) = s$.

The value of $\mathbf{M}(1,2:n)$ can straightforwardly be determined based on the equality constraints in the primal problem. The final solution is thus given by: $\mathbf{M}(1,1) = s(q-1) = \infty$, $\mathbf{M}(2,2) = s$ and $\mathbf{M}(2:n,1)$ as determined by the equality constraints of the primal problem. If we define $\mathbf{m} \in \Re^n$ as $\mathbf{m} = \frac{1}{\sqrt{s}}\mathbf{V}(:,2:n)\mathbf{M}(2:n,1)$ (satisfying $\mathbf{m}'\mathbf{d} = 0$), we can state this result conveniently in terms of the original variables:

$$\begin{array}{rcl} \widehat{\boldsymbol{\Gamma}} & = & s\mathbf{V}(:,2)\mathbf{V}(:,2)' + (q-1)\mathbf{1}\mathbf{1}' + \mathbf{m}\mathbf{1}' + \mathbf{1}\mathbf{m}', \\ & = & s\widetilde{\mathbf{y}}\widetilde{\mathbf{y}}' + (q-1)\mathbf{1}\mathbf{1}' + \mathbf{m}\mathbf{1}' + \mathbf{1}\mathbf{m}'. \end{array}$$

with $q \to \infty$. ■

References

M. F. Anjos and H. Wolkowicz. Strengthened semidefinite relaxations via a second lifting for the MAX-CUT problem. *Discrete Applied Mathematics*, 119:79–106, 2002.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. of the 18th International Conf. on Machine Learning (ICML)*, 2001.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, U.K., 2004.

S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (series B)*, 95(2):329–357, 2003.

S. Burer and R. D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming (series A)*, 103(3):427–444, 2005.

O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA, 2003.

N. Cristianini, J. Shawe-Taylor, and J. Kandola. Spectral kernel methods for clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

T. De Bie and N. Cristianini. Convex methods for transduction. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004a.

T. De Bie and N. Cristianini. Kernel methods for exploratory data analysis: a demonstration on text data. In *Proceedings of the International Workshop on Statistical Pattern Recognition (SPR2004)*. Lisbon, Portugal, August 2004b.

T. De Bie, J. Suykens, and B. De Moor. Learning from general label constraints. In *Proceedings of IAPR International Workshop on Statistical Pattern Recognition (SPR)*. 2004.

M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42 (6):1115–1145, 1995.

C. Helmberg. Semidefinite programming for combinatorial optimization. Habilitationsschrift ZIB-Report ZR-00-34, TU Berlin, Konrad-Zuse-Zentrum Berlin, 2000.

C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10:673–696, 2000.

T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *IJCAI*, 2003.

K. Lang. Finding good nearly balanced cuts in power law graphs. Technical Report YRL-2004-036, Yahoo! Research, 2004.

A. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software, Special issue on Interior Point Methods (CD supplement with software)*, 11-12:625–653, 1999.

L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

E. P. Xing and M. I. Jordan. On semidefinite relaxation for normalized k-cut and connections to spectral clustering. Technical Report CSD-03-1265, Division of Computer Science, University of California, Berkeley, 2003.