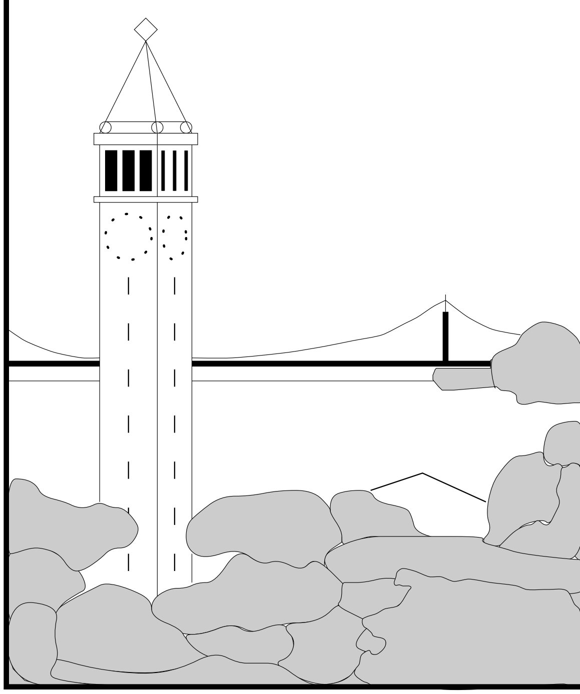
On semidefinite relaxation for normalized k-cut and connections to spectral clustering

Eric P. Xing
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
`epxing@cs.berkeley.edu`

Michael I. Jordan
Computer Science and Statistics
University of California, Berkeley
Berkeley, CA 94720
`jordan@cs.berkeley.edu`



Report No. UCB/CSD-3-1265

June 2003

Computer Science Division (EECS)
University of California
Berkeley, California 94720

On semidefinite relaxation for normalized k-cut and connections to spectral clustering

Eric P. Xing
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
epxing@cs.berkeley.edu

Michael I. Jordan
Computer Science and Statistics
University of California, Berkeley
Berkeley, CA 94720
jordan@cs.berkeley.edu

June 2003

Abstract

The normalized cut (NC) provides a plausible cost function for clustering. Finding an optimal NC is NP-hard, and the well-known spectral graph partition methods rely on a loose *spectral relaxation*. In this paper, we study a semidefinite programming (SDP) model for the normalized k-cut (k-NC), a generalization of the conventional NC on bisection to k-section that naturally relates to multi-way clustering. Our *SDP relaxation* to k-NC provides a tighter lower bound on the cut weight, as well as a better feasible cut, than that of the spectral relaxation. In applications to clustering, however, the improved solution to k-NC does not translate into improvements in clustering—the results for the SDP approach and spectral relaxations are very similar. We conclude that the normalized cut criterion is useful in terms of leading via various relaxations to reasonable clustering methods, but normalized cut alone does not characterize optimal clusterings.

1 Introduction

Clustering and segmentation problems are intrinsically hard to formulate. Classical algorithms such as K-means or EM optimize simple objectives, such as (minimizing) the spread over centroids or (maximizing) the log likelihood under a mixture model. While these criteria are seductively simple to define, and bear intuitive interpretations, they can give poor solutions due to their simplifying assumptions about the cluster structure; e.g., that each cluster constitutes a "densely populated" convex sub-manifold, or that the density of each cluster is Gaussian, etc., assumptions which are often not warranted in real data (see [10] for a gallery of problems that do not satisfy these assumptions). Furthermore, these cost functions are in general not convex.

Spectral clustering (SC) algorithms [7, 10] provide an alternative approach to clustering which have had significant practical successes [12, 17]. SC generally makes use of the top eigenvectors of an affinity matrix to transform the original clustering problem into one in a lower-dimensional eigenspace. While this procedure can be interpreted as an optimization procedure, it is generally not the case that the cost function bears a direct relationship to the clustering problem. Thus comparisons between SC and the centroid-based algorithms tend to be somewhat indirect.

There is an analogy between spectral clustering and the *normalized cut* (NC) problem [2, 12], a graph-theoretic problem of bisection. This problem is specified in terms of a ratio between intra-cluster and inter-cluster similarities of vertices, and this would seem to have a natural relationship to data clustering. Is NC the right criterion function for clustering?

In this paper, we build upon recent work of Gu et. al. [5], who pointed out a connection between SC and the *normalized k-cut* (k-NC) of a graph, a generalization of conventional NC on bisection to k-section. They formulate a graph-theoretic model for general k-way clustering based on optimizing the normalized k-cut, which amounts to a constrained quadratic optimization problem for which an SC algorithm (e.g., as described in [10]) can be thought of as a spectral relaxation. Although it is generally believed that spectral relaxations work well in practice, Guattery and Miller [6] present graphs for which spectral partitioning yields poor separators. Motivated by recent success of semidefinite programming (SDP) as a powerful tool for approximating difficult graph optimization problems such as Max-Cut and graph partitioning [4, 3, 8], we have developed an SDP relaxation for normalized k-cut, which yields a tighter relaxation of k-NC than that of a spectral relaxation. With this improved tool in hand, we can then ask whether improvement in normalized cuts yield improvements in clustering.

2 Preliminaries: graph partition

Let $G(V, E, A)$ be a weighted undirected graph with nodes $V = \{1, \ldots, n\}$, edges E and nonnegative weights a_{ij}, for $(i,j) \in E$ ($a_{ij} = 0$ if there is no edge between node i and j; also $a_{ii} = 0, \forall i$). We refer to the symmetric matrix $A = \{a_{ij}\}$ as the *affinity matrix*. We equip the space of $n \times n$ matrices with the trace inner product $A \bullet B = \text{tr } AB$; let $A \succeq 0$ denote positive semidefiniteness ($A \succeq B$ denotes $A - B \succeq 0$); and let $A \geq 0$ denote elementwise non-negativity of A. The linear operator $\text{Diag}(a)$ forms a diagonal matrix from the vector a, and its adjoint operator $\text{diag}(A)$ yields a vector containing the diagonal elements of A. We denote by e_k the vector containing k ones.

A classical graph partition (GP) problem involves partitioning the nodes into k disjoint subsets $(S_1, \ldots, S_k)$ of specified sizes $m_1 \geq m_2 \geq \ldots \geq m_k$, $\sum_{j=1}^{k} m_j = n$, so as to minimize the total weight of the edges connecting nodes in distinct subsets of the partition. This problem is well known to be NP hard.

A k-way node partition can be represented by an *indicator matrix* $X \in \mathbb{R}^{n \times k}$ with the columns, $x_j = (x_{1j} \;\; x_{2j} \ldots x_{nj})^t$, being the *indicator vector* for the set S_j, $\forall j$:

$$x_{ij} = \begin{cases} 1 & : \quad \text{if } \; i \in S_j \\ 0 & : \quad \text{if } \; i \notin S_j \end{cases}. \tag{1}$$

For each partition X, the total weight of the edges connecting nodes within cluster S_i to its complement $\bar{S}_i$ is, $w(S_i, \bar{S}_i) = \frac{1}{2} x_i^t (D - A) x_i$, where $D = \text{Diag}(Ae_n)$ is the *weighted degree matrix*. The total cut of a k-way partition is therefore $C_k = \sum_i \frac{1}{2} x_i^t (D - A) x_i = \frac{1}{2} \text{tr} X^t L X$, where $L \triangleq D - A$ is the *Laplacian matrix* associated with G. Thus classical GP amounts to minimizing C_k subject to constraint (1).

3 Clustering and graph partition: the normalized cut problem

A pairwise metric-based clustering problem can be formulated as a graph partition problem. Let $V(G)$ map to an n-item dataset, with $A = (a_{ij})$ the *affinity matrix*, i.e., a_{ij}, $i \neq j$, $i, j = 1, \ldots, n$, encode a nonnegative "similarity measure" for points i and j. A legitimate objective of data clustering is to partition the data such that the sum of pairwise affinities of points from different clusters is minimized. But to prevent trivial solutions, we "normalize" the cut of each cluster with the total weighted degree of nodes in this cluster, such that the weight of a normalized k-cut can be defined as:

$$\begin{aligned} NC_k &= \sum_i \frac{w(S_i, \bar{S}_i)}{w(S_i, V)} = \frac{1}{2} \sum_i \frac{x_i^t (D - A) x_i}{x_i^t D x_i} \\ &= \frac{1}{2} \text{tr Diag}((x_1^t D x_1)^{-1}, \ldots, (x_k^t D x_k)^{-1})(X^t (D - A) X) \\ &= \frac{1}{2} \text{tr } [(X^t D X)^{-1} (X^t L X)]. \end{aligned} \tag{2}$$

Note that from the 2nd to the 3rd line of the above equations, we use the fact that X^tDX is a diagonal matrix whose inversion is just the elementwise inversion.

According to Eq. 1, k-way clusterings are in one-to-one correspondence with the set

$$\mathcal{F}_k = \{X : Xe_k = e_n,\ X^te_n \geq c,\ x_{ij} = \{0, 1\}\},$$

where c is the minimum size of each cluster. Thus, a clustering problem based on k-NC can be modeled as the following optimization problem

$$NC_k^* = \frac{1}{2} \min_{X \in \mathcal{F}_k} \mathrm{tr}(X^tDX)^{-1}(X^tLX). \tag{NC}$$

NC is NP-hard even for $k = 2$ [12]. The difficulty lies in the nonlinear structure of the objective, and the combinatorial nature of the feasible set.

Generalizing the probabilistic analysis by Melia and Shi on 2-NC [9], we can interpret the weight of the k-NC as

$$NC_k = \sum_i P(S_i \to \bar{S}_i | S_i), \tag{3}$$

which corresponds to the total conditional probabilities of escaping from each of the clusters via a single random walk started within the respective clusters. Thus k-NC directly relates to the concept of low conductivity sets and mixing time of Markov random walk, suggesting that the k-NC criteria is well founded for clustering.

3.1 k-NC as a quadratic optimization problem

Defining $S = \mathrm{Diag}(s_1, \ldots, s_k) = (X^tDX)^{1/2}$, $Y = D^{1/2}XS^{-1}$, $W = D^{-1/2}AD^{-1/2}$, the following optimization problem is equivalent to k-NC, with a somewhat simpler objective function.

$$(\mathrm{P}) \quad \left\{ \begin{array}{lll} \underline{\max} & \frac{1}{2}\mathrm{tr}Y^tWY & \\ \underline{\text{s.t.}} & Y^tY = I_k & (a) \\ & (D^{-1/2}y_j)_i \in \{0, s_j^{-1}\}, \forall i, j & (b) \\ & YSe_k = \mathrm{diag}(D^{1/2}) & (c) \\ & SY^t\mathrm{diag}(D^{1/2}) \geq c & (d) \\ & S = \mathrm{Diag}(s_1, \ldots, s_k) \in \mathbb{R}^n_+ & (e) \end{array} \right.$$

Constraint (a) is due to our definition of Y; (b) is to ensure that $D^{-1/2}y_j$ is a column vector whose elements can take only two discrete values (zero or a positive constant originating from the binary assignment of X); (c) corresponds to the membership exclusivity condition for X in $\mathcal{F}_k$; (d) corresponds to the minimum size condition of each cluster in $\mathcal{F}_k$; and (e) comes from the definition of the normalization matrix. Note that this is not a *convex* optimization problem because W may not be positive semidefinite.

3.2 Spectral clustering and k-way normalized cut

One form of the k-way spectral clustering algorithm, described in [10], uses the truncated eigenvector basis of W (top k eigenvectors) to re-represent the original data (known as *embedding*) for subsequent clustering using standard methods such as K-means. Although one may simply view the embedding step as the best rank k approximation of W in terms of squared error, perhaps a better way to justify the use of the W matrix is via its connection to the k-NC.

Consider the following weak relaxation of (P), which drops all but the first constraint,

$$(\mathrm{P0}) \quad \left\{ \begin{array}{ll} \underline{\max} & \frac{1}{2}\mathrm{tr}Y^tWY \\ \underline{\text{s.t.}} & Y^tY = I_k. \end{array} \right.$$

Clearly, the optimal solution of (P0) is attained at $Y = U_k$, whose columns $u_i,\ i = 1, \ldots, k$ are eigenvectors corresponding to the (ordered) top k largest eigenvalues of W, and $NC_k^*(\text{P0}) := k - \sum_{i=1} \lambda_k(W)$. This is exactly what is achieved by the embedding procedure in [10]. Thus this form of SC can be regarded as solving a *spectral relaxation* of (P). The subsequent unit-ball projection of the embedded points and the K-means procedure can be regarded as recovering the feasible solution of the k-NC problem, but with a somewhat indirect cost function—the distortion measure of the points in the k-eigenfeature space.

4 Semidefinite relaxations for NC

Although the eigenvalue relaxation (P0) yields a globally optimum approximation to k-NC, it corresponds to a rather loose relaxation. In this section we seek a tighter relaxation using semidefinite programming (SDP).

4.1 Semidefinite programming

Semi-definite programming (SDP) refers to the problem of optimizing a convex function over the convex cone of symmetric and positive semidefinite matrices, subject to linear equality constraints [14]. A canonical (primal) SDP has the form:

$$(\mathbf{SDP}) \qquad \begin{cases} \underline{\min} & C \bullet X \\ \underline{\text{s.t.}} & A_i \bullet X = b_i \quad \text{for } i = 1, \ldots, m \\ & X \succeq 0 \end{cases}$$

Because of the convexity of the objective function and the feasible space, SDP problems have a single global optimum. With the development of efficient, general purpose solvers based on interior-point methods (e.g., Sedumi [13]), SDP has become a powerful tool in solving difficult combinatorial optimization problems.

4.2 SDP relaxation

We now derive the semidefinite model for the normalized k-cut problem, following a strategy similar to that described in [8]. The basic idea is to linearize tr Y^tWY by tr WZ, where Z corresponds to YY^t. Let us define the set $\mathcal{T}_k$:

$$\mathcal{T}_k := \{Z : \ \exists X \in \mathcal{F}_k \text{ such that } \ Z = YY^t, \text{where } Y = D^{1/2}X(X^TDX)^{-1/2}\}.$$

Thus P reads

$$NC_k^* = \max\{\frac{1}{2}\text{tr } WZ \ : \ Z \in conv(\mathcal{T}_k)\}.$$

Note that due to linearization of the objective, our feasible set can be rewritten as the convex hull of the original set $\mathcal{T}_k$. One of the difficulties of this optimization problem is approximating the convex hull of $\mathcal{T}_k$ by outer approximations that can be handled efficiently. To derive these relaxations, we introduce the following sets, all of which contain $\mathcal{T}_k$.

First, note that since $Y^tY = I_k$, we can optimize over

$$\mathcal{O}_k := \text{conv}\{YY^t \ : \ Y^tY = I_k\}.$$

The following lemma provides an equivalent description of this set in terms of Z.

Lemma 1 (Overton and Womersley [11])

$$\mathcal{O}_k = \{Z \ : Z = Z^t, \text{tr } Z = k, \ I \succeq Z \succeq 0\}.$$

This orthonormal outer approximation is not directly useful for the general classical GP problem as described in §2, because for a general GP, the indicator matrix X is not orthonormal due to the size constraint, i.e, $X^tX = \text{Diag}(m_1, \ldots, m_k) \equiv M$. But for a special case of GP, the equi-partition, where $M = mI$, $\mathcal{O}_k$ can be used directly as a constraint set for, say, $\frac{1}{m^{1/2}}X$, which is widely used for SDP relaxation of graph equi-partition [8].

The following lemma leads to a second outer approximation.

Lemma 2 $Z = YY^t$ satisfies transportation constraints: $Z\ \text{diag}(D^{1/2}) = \text{diag}(D^{1/2})$.

Proof. According to the definition of Y,

$$\begin{aligned} YY^t\text{diag}(D^{1/2}) &= (\sum D^{1/2}X(X^tDX)^{-1/2}(X^tDX)^{-1/2}X^tD^{1/2}D^{1/2}e_n \\ &= D^{1/2}X(X^tDX)^{-1}X^t\text{diag}(D) \\ &= D^{1/2}X(X^tDX)^{-1}[x_1^tDx_1,\ldots,x_k^tDx_k]^t \\ &= D^{1/2}X(X^tDX)^{-1}(X^tDX)e_k \\ &= D^{1/2}Xe_k \\ &= D^{1/2}e_n = \text{diag}(D^{1/2}) \end{aligned}$$

Note that from the 2nd to the 3rd line, and from the 3rd to the 4th line, the equalities can be varified using the properties of X as a cluster indicator matrix (i.e. Eq. (1)). ∎

Thus, we have the following linear submanifold of matrices that contains $\mathcal{T}_k$,

$$\mathcal{E} := \{Z\ :\ Z\ \text{diag}(D^{1/2}) = \text{diag}(D^{1/2})\}.$$

This linear constraint is unique to k-NC, implicitly capturing the fact that Y is a "normalized" indicator matrix even though we do not explicitly know the value of the normalization factor (i.e., $(X^tDX)^{-1}$). A simple counterpart in the equipartition case is $Xe_n = me_n$ [8], but for general GP, no obvious analogy can be drawn.

Finally, since $X \geq 0$, $D > 0$, it is obvious that $\mathcal{T}_k$ is contained in the nonnegative orthant

$$\mathcal{N} := \{Z\ :\ Z \geq 0\}.$$

From Lemma 1, the eigenvalue bound attained by SC in (P0) can be reformulated as a SDP relaxation of the k-NC problem as follows.

Theorem 3 Eigenvalue bound of k-NC is an SDP relaxation

$$\begin{aligned} NC_k(P0) &= k - \max\{\frac{1}{2}\text{tr}\ WZ\ :\ Z \in \mathcal{O}_k\} \\ &= \min\{\frac{1}{2}\text{tr}\ \bar{L}Z\ :\ Z \in \mathcal{O}_k,\ \bar{L} = I_k - W\} \end{aligned} \tag{4}$$

This formulation appears very similar to an SDP rendition of the Donath-Hoffman bound for graph equipartition [8]. But note that for k-NC, what enters the cost function is the "rescaled" Laplacian $\bar{L}$, rather than the original graph Laplacian.

Imposing the additional conditions $Z \in \mathcal{E}$ and $Z \in \mathcal{N}$ still leads to a tractable relaxation, but will give a stronger bound. Thus we arrive at the following strengthened SDP relaxation of the k-NC problem.

Theorem 4 For a clustering problem defined by (G, A), we have $NC_k(\text{P0}) \leq NC_k(\text{P1}) \leq NC_k^*$, where $NC_k(\text{P1})$ is defined as

$$(\text{P1}) \quad \left\{ \begin{array}{lll} \underline{\min} & \frac{1}{2}\text{tr}\ \bar{L}Z & \\ \underline{\text{s.t.}} & Z\ \text{diag}(D^{1/2}) = \text{diag}(D^{1/2}) & (1) \\ & Z \geq 0 \quad \text{elementwise} & (2) \\ & \text{tr}\ Z = k & (3) \\ & Z \succeq 0,\ Z = Z^t & (4) \\ & I - Z \succeq 0 & (5) \end{array} \right.$$

Comparing to the SDP relaxations of the classical GP or Max-Cut problem [4, 8], constraints (1), (3) and (5) are specific to k-NC. (P1) can be straightforwardly cast into the canonical SDP format of a standard SDP solver such as Sedumi.

5 Finding a Closest Feasible Solution

Due to the relaxation, the optimal solution of problems (P0) or (P1) are in general not feasible for (P). Thus we need to recover from the approximate solution a closest feasible solution, X, to the original k-NC problem. This process is often referred to as *rounding.* We use the following rounding scheme in this paper.

- From $Z = Y'Y'^t$, obtain Y' via SVD (Y' is usually full rank rather than rank k).
- Based on condition (b) in (P), rescale Y': $Y'' = D^{-1/2}Y'$.
- Treat each row in Y'' as a point in $\mathbb{R}^n$; clustering them with restarts using any standard algorithm (e.g., K-means); pick the X with the lowest NC_k value.

This rounding scheme, which we refer to as "rank-n KM rounding" (and "rank-k rounding" if only k eigenvectors of Z are used), is related to the randomized projection heuristic studied by [4] in their work on Max-Cut. In this approach, the label (-1 or +1) of each vector is chosen according to whether the vector is above or below a randomly chosen hyperplane passing through the origin. In [3] this scheme is generalized to max k-cut.

A majority of the spectral clustering methods adopt a K-means clustering procedure on row-rescaled, top k eigenvectors in Y derived from (P0). In particular, the rescaling projects all row-associated points onto a k-dimensional unit ball, therefore the K-means actually finds clustering with minimal intra-cluster angular spread. This rounding scheme distinguishes SC from the k-NC problem because a different cost function (i.e., not NC_k) is used to determine the optimal rounding. As a result, despite the similarity of SC and NC, they may be aiming at capturing different objectives.

Recently, [1] showed that the distortion measure of a *weighted* K-means applied on the relaxed solution matrix Y' corresponds to a difference measure between the subspaces spanned by the relaxed solution Y' and a feasible solution $Y = [\frac{D^{1/2}x_1}{x_1^t D x_1}, \ldots, \frac{D^{1/2}x_k}{x_k^t D x_k}]$, where $X = [x_1, \ldots, x_k]$ is a feasible indicator matrix obtained via K-means from Y' [1]:

$$\begin{aligned} \epsilon &= \frac{1}{2}\|Y'Y'^t - YY^t\|_F^2 \\ &= \min_{[\mu_1,\ldots,\mu_k]} \sum_j \sum_i d_i \|d_i^{1/2} y'_i - \mu_j\|^2, \end{aligned}$$

where d_i denotes the ith diagonal element of D. This cost function for K-means is similar to the one we use, except that each *"point-to-centroid"* distance is now weighted by the *"degree"* of the point, d_i. Note that although the foregoing ϵ defines a good measure of rounding error, it is not necessarily coincide with minimal deviation from optimal normalized cut, but nevertheless provide a good heuristic of moving close to it.

6 Experimental results

In this section, we study normalized k-cut defined on high-dimensional real datasets of various natures, whose cluster structures are not directly visible and likely to be ambiguous. Human labels are available for all dataset, and we use them as a gold standard for clustering. We collected six real datasets, two from the Whitehead Institute microarray database of cancer samples (the "Lung Carcinomas" set and the "multi-cancer dataset"), two from the Reuters-21578 collection (subsampled from the four smallest, and four largest categories, respectively, with feature vectors based on word counts), a protein dataset from http://www.nersc.gov/∼cding/protein/, and the soybean dataset from the UCI repository. The first four datasets are very high-dimensional (>5000), and we use a simple unsupervised filtering procedure described in [15] to reduce the total features to 1000.

[1]Because Y' is defined up to a rotation matrix, a common difference measure is thus to compare the orthogonal projection operators on the subspaces, i.e., the Frobenius norm between $Y'Y'^t$ and $Y_f Y_f^t$.

6.1 Optimizing normalized k-cut

We generate k-NC problems by constructing the affinity matrix A using a Gaussian kernel, $a_{ij} = \exp\{-d_{ij}^2/\sigma\}$, where d_{ij} is Euclidean distance. We let $\sigma = r \cdot c$, where $c = \frac{1}{n}\sum_i \min_j d_{ij}$ is the mean nearest-neighbor distance, and r is an integer called the *affinity window*. It is easier empirically to choose r (than σ), as it roughly translates to a "sensitivity range" of neighboring points [2].

Table 1 summarizes the results of k-NC on the six datasets using SDP and spectral relaxations. We show the lower bounds of NC_k^* computed from both relaxations, as well as the NC_k attained by the feasible solutions yielded by the SDP and the spectral methods, with several different rounding schemes. Due to the NP-hardness of the problems, we can not compute the NC_k^* of graphs for evaluation, but the ratio f/b between the feasible cut and the lower bound straddles the optimal value. Consistent with Theorem 4, the SDP bound is tighter than the spectral bound, and except for random-projection rounding, SDP almost always leads to better (tie for soybean) feasible NC_k values than that of the spectral methods. For SDP relaxation, the K-means-based roundings are far better than the random-projection rounding, and it seems that rounding using the rank-n relaxed solution is slightly better than that of rank-k.

For spectral relaxation, even when we round to the best feasible NC_k after K-means, the result is not much better than the one attained by a standard SC methods, suggesting that the SC algorithm is in fact very close to solving a k-NC with spectral relaxation.

Table 1: Performance on k-NC. (KM: K-means; rp: random projection; SC: spectral clustering)

				SDP relaxation							Spectral relaxation				
					rank-n rp rounding		rank-n KM rounding		rank-k KM rounding			rank-k KM rounding		SC	
dataset	size	k	r	lower-b	feas. X	f/b	feas. X	f/b	feas. X	f/b	lower-b	feas. X	f/b	feas. X	f/b
cancer1	120	4	1	1.923	2.070	1.077	1.960	1.019	1.961	1.020	1.826	2.024	1.108	2.032	1.113
cancer2	122	12	2	6.955	9.887	1.422	7.820	1.124	7.905	1.137	6.279	8.316	1.324	8.456	1.347
reuters1	111	4	2	2.667	2.725	1.022	2.681	1.005	2.683	1.006	2.597	2.705	1.042	2.705	1.042
reuters2	120	4	1	1.850	1.908	1.031	1.860	1.005	1.860	1.005	1.793	1.900	1.060	1.900	1.060
protein	116	6	2	4.408	4.625	1.049	4.466	1.013	4.491	1.019	4.240	4.499	1.061	4.519	1.066
soybean	47	4	10	0.112	0.141	1.259	0.121	1.080	0.121	1.080	0.100	0.121	1.210	0.121	1.210

6.2 Clustering based on normalized k-cut

Now we examine whether the feasible solutions to normalized k-cut found by relaxed optimization agree well with the cluster labels of the datapoints, and whether a better value of NC_k suggests better agreement. Table 2 summarizes the consistency measures (as defined in [16]) between the feasible X and the sample labels for the six datasets. We also give the K-means results for reference. Comparing the rank-n KM rounding for SDP and the SC column, we see that the results are quite similar, with SDP slightly better in some cases, with SC slightly better in others, and with parity in several cases. Overall, we do not find that a better value of NC_k translates into a higher consistency with the label.

Table 2: Performance for data clustering.

	SDP relaxation			Spectral relaxation		K-means
dataset	rank-n rp rounding	rank-n KM rounding	rank-k KM rounding	rank-k KM rounding	SC	
cancer1	0.5213	0.7558	0.7465	0.7962	0.7959	0.7752
cancer2	0.5159	0.7548	0.7305	0.7309	0.7112	0.6608
reuters1	0.6785	0.6200	0.6346	0.6272	0.6272	0.6189
reuters2	0.6799	0.7016	0.7016	0.6693	0.6693	0.6829
protein	0.5440	0.6530	0.6404	0.6291	0.6459	0.6150
soybean	0.5681	0.9014	0.9014	0.9014	0.9014	0.7910

[2] In the following experiment, we pick r based on multiple trials that finds a range that gives stable feasible solutions.

7 Discussion

We have presented an SDP relaxation for the normalized k-cut problem. We show that the SDP relaxation gives a tighter lower bound in theory than that of spectral relaxation; moreover, this is borne out in practice. Modulo computational issues, the SDP relaxation may prove to be a useful approach to solving normalized k-cut problems.

We do not find a compelling relationship between improvement in normalized k-cut values, and improvement in clustering performance. This shows that although both the SDP approach and the SC approach can be viewed as relaxations of normalized k-cut, their clustering performance is not explained entirely in terms of their performance on the partitioning problem. Thus, while the normalized k-cut criterion is useful in terms of formulating clustering problems as optimization problems, there may be other criteria that more directly capture the underlying clustering criterion of spectral clustering and explain its success.

A promising application of the SDP formulation of the normalized k-cut problem is that it provides a principle framework for incorporating side-information (e.g., small amount of (dis)similarity constraints as studied in [16]) during optimization to achieve a transduction effect, which is not directly achievable in the spectral clustering paradigm [3]. Currently, we are exploring this direction, and its combination with the metric learning approach.

Acknowledgements

We thank Francis Bach for help discussions on the mathematical interpretations of the K-means rounding.

References

[1] F. R. Bach and M. I. Jordan. Learning spectral clustering. TR CSD-03-1249, CS Division, UC Berkeley, 2003.

[2] F. Chung. *Spectral Graph Theory.* No. 92 in CBMS Regional Conference Series in Mathematics, American Mathematical Society, 1997.

[3] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. In *Integer Programming and Combinatorial Optimization*. Springer, 1995.

[4] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiabil ity problems using semidefinite programming. *JACM*, 42:1115–1145, 1995.

[5] M. Gu, H. Zha, C. Ding, X. He, and H. Simon. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering. TR CSE-01-007, Penn State University, 2001.

[6] S. Guattery and G. L. Miller. On the quality of spectral separators. *SIAM Journal on Matrix Analysis and Applications*, 19(3):701–719, 1998.

[7] R. Kannan, S. Vempala, and A. Vetta. On clusterings – good, bad and spectral. In *Proc. of 41st annual Symposium on Foundations of Computer Science*, 2001.

[8] S. E. Karisch and F. Rendl. Semidefinite programming and graph equipartition. In *Topics in Semidefinite and Interior-Point Methods*, volume 18, pages 77–95. AMS, 1998.

[9] M. Maila and J. Shi. A random walks view of spectral segmentation. In *AISTATS*, 2001.

[10] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[3] Although one could resort to an "indirect" approach, i.e., using the side-information to learn a metric [16], select features, or fit the scaling constant [1] for the distance measure, and "prepossess" the data based on these results before conducting the spectral clustering.

[11] M. Overton and R. Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM J. on Matrix Analysis and Applications*, 13:41–45, 1992.

[12] J. Shi and J Malik. Normalized cuts and image segmentation. In *PAMI*, 2000.

[13] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

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

[15] E. P. Xing and R. M. Karp. Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. In *ISMB*, 2001.

[16] E. P. Xing, A. Y. Ng, M. I. Jordan, , and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2002.

[17] S. Yu, R. Gross, and J. Shi. Concurrent object segmentation and recognition with graph partitioning. In *NIPS*, 2002.