McMaster University

Advanced Optimization Laboratory

Title:

0-1 Semidefinite Programming for Spectral Clustering: Modeling and Approximation

Author:

Jiming Peng

AdvOl-Report No. 12, 2005

July 2005, Hamilton, Ontario, Canada

0-1 Semidefinite Programming for Spectral Clustering: Modeling and Approximation

Jiming Peng [*]

July 18, 2005

Abstract

In the past few years, spectral clustering has caught much attention in the machine learning community and numerous algorithms have been proposed and well-studied. In this paper, we present a unified framework for these methods based on a new optimization model: 0-1 semidefinite programming (0-1 SDP). We also show how the so-called balanced clustering can be embedded into the new 0-1 SDP model. Secondly, we consider the issue of how to solve the underlying 0-1 SDP problem. We consider two approximation methods based on principal component analysis (PCA) and projected PCA, respectively and prove that both algorithms can provide a 2-approximation to the original clustering problem. The complexity of these approximation algorithms are also discussed.

Key words. K-means clustering, Spectral clustering, Principal component analysis, Semidefinite programming, Approximation.

1 Introduction

In general, clustering involves partition a given data set into subsets based on the closeness or similarity among the data. Clustering is one of major issues in data mining and machine learning with many applications arising

[*] Advanced optimization Lab, Department of Computing and Software McMaster University, Hamilton, Ontario L8S 4K1, Canada. Email: pengj@mcmaster.ca. This work was supported by the NSERC discovery grant # RPG 249635-02 of Canada and a PREA award in Ontario. It is also supported by the MITACS project "New Interior Point Methods and Software for Convex Conic-Linear Optimization and Their Application to Solve VLSI Circuit Layout Problems".

from different disciplines including text retrieval, pattern recognition and web mining[11, 14]. There are many kinds of clustering problems and algorithms, resulting from various choices of measurements used in the model to measure the similarity/dissimilarity among entities in a data set. For a comprehensive introduction to the topic, we refer to the book [11, 14], and for more recent results, see survey papers [7] and [12].

In the present paper, we are mainly concerned about the so-called spectral clustering. Given a set of points $\mathcal{X} = \{x_i \in \Re^m : i = 1, \cdots, n\}$, in spectral clustering we first define a similarity matrix $W = [w_{ij}]$ where $s_{ij} = \phi(x_i, x_j)$ for some kernel function $\phi(\cdot)$, which can be further interpreted as the weight of the edge (x_i, x_j) in a graph with vertex set $\mathcal{X}$. We then solve a graph partitioning problem or an optimization problem to cluster the data set.

There are many variants of spectral clustering depending on various choices of the similarity matrix W and the optimization model we solve [21, 1, 26, 18]. Among others, we shall focus on the following three specific clustering algorithms: (1) The so-called normalized cut for image segmentation introduced by Shi and Malik [21] and later investigated by Xing and Jordan [26]; (2) The classical K-means clustering based on minimum sum of squared errors [16]; (3) The kernel (weighted) K-means by Dhillon, Guan and Kulis [5]. We point out that the interrelations among these algorithms has been explored in [5]. In[23], Verma and Meila compared several clustering algorithms that included the above mentioned methods and introduced the concept of the so-called perfect set for which the algorithm works well. The main purpose of this work is to present a unified framework for the above-mentioned methods and investigate how to solve the unified optimization model. However, as we shall see later, our unified framework covers not only these few algorithms, but also some other clustering scenarios as well.

This paper is inspired by our recent paper [20] where we reformulated the classical K-means clustering as the so-called 0-1 SDP and proposed a 2-approximation method based on projected PCA to attack the underlying 0-1 SDP model. It has been observed that except for its excellent theoretical properties and high computational efficiency, the algorithm in [20] can be easily extended to deal with the so-called balanced clustering where the cardinality of the clusters is bounded.

A popular approach for spectral clustering is to use PCA to reduce the dimension of the data set and then perform the clustering in the reduced space [4, 6, 27, 18]. There are several issues need to be addressed in such an approach. First, how the new clustering problem in the reduced space can be solved? Secondly, how to estimate the quality of the solution obtained from the reduced problem. Unfortunately, most of the above mentioned works did not provide a very satisfactory answer to these questions and only Drineas et'al [6] showed that their algorithm can give a 2-approximation to the original clustering problem. The algorithm in [20] follows a similar vein as the algorithm in [6] in the sense that both algorithms use PCA to reduce the dimension of the space and both algorithms can provide a 2-approximation solution to original problem. However, the motivation behind these algorithm are quite different. In [6], PCA is employed to reduce the dimension of data so that the reduced problem can be solved relatively easily, while the algorithm in [20] is based on the relaxation of the 0-1 SDP model. Note that there are several different ways to relax the 0-1 SDP model. On the other hand, the dimension of the working space in [20] is smaller than that in [6]. This simplifies the algorithm and improves its efficiency.

There have been several works on the SDP relaxation for spectral clustering. For example, Zha et'al [27] rewrote the objective in the classical K-means clustering as a convex quadratic function and then relaxed it an SDP problem, which can be solved by using singular values decomposition of the underlying coefficient matrix. Similar discussions for normalized cuts and kernel K-means can be found in [26]. However, different from the above-mentioned approaches, we elaborate more on how to characterize spectral clustering precisely by means of matrix arguments. In particular, we show that all the three methods can be embedded into the so-called 0-1 semidefinite programming (SDP), which can be further relaxed to polynomially solvable linear programming (LP) and SDP. Our model not only provides novel avenues for spectral clustering, but also gives insightful analysis about these algorithms. For example, as a byproduct of our analysis, we show that the algorithm proposed by Shi and Malik [21] can provide a 2-approximation to the original bi-clustering problem.

The paper is organized as follows. In Section 2, we show that all the three methods for spectral clustering can be modelled as 0-1 SDP, which allows convex relaxation such as SDP and LP. In Section 3, we consider two approximation algorithms for solving our 0-1 SDP model. The first is the popular approach for spectral clustering that uses PCA to reduce the dimension of the data set, and then performs the clustering task in the lower dimension. The second one is based on the projected PCA which can be viewed as a slight improvement of the first algorithm. Approximate ratios for both algorithms between the obtained solution and the global solution of the original spectral clustering are estimated. Finally we close the paper by few concluding remarks.

2 0-1 SDP for spectral clustering

In this section, we present a unified framework for spectral clustering under the umbrella of 0-1 SDP model. The section has three parts. In the first part, we briefly describe SDP and 0-1 SDP. In the second part, we review the 0-1 SDP model for the classical K-means clustering and kernel K-means. In the last subsection, we establish the equivalence between 0-1 SDP and normalized cuts.

2.1 0-1 Semidefinite Programming

In general, SDP refers to the problem of minimizing (or maximizing) a linear function over the intersection of a polyhedron and the cone of symmetric and positive semidefinite matrices. The canonical SDP takes the following form

$$\textbf{(SDP)}\begin{cases} \min & \mathrm{Tr(WZ)} \\ S.T. & \mathrm{Tr(B_iZ) = b_i \quad fori = 1, \cdots, m} \\ & Z \succeq 0 \end{cases}$$

Here Tr(.) denotes the trace of the matrix, and $Z \succeq 0$ means that Z is positive semidefinite. If we replace the constraint $Z \succeq 0$ by the requirement that $Z^2 = Z$, then we end up with the following problem

$$\textbf{(0-1 SDP)}\begin{cases} \min & \mathrm{Tr(WZ)} \\ S.T. & \mathrm{Tr(B_iZ) = b_i \quad fori = 1, \cdots, m} \\ & Z^2 = Z, Z = Z^T \end{cases}$$

We call it 0-1 SDP owing to the similarity of the constraint $Z^2 = Z$ to the classical 0-1 requirement in integer programming.

2.2 0-1 SDP Model for K-means and Kernel K-means clustering

Let us consider the following special form of 0-1 SDP.

$$\begin{aligned} \min \quad & \mathrm{Tr(W(I - Z))} \\ & Ze = e, \mathrm{Tr(Z) = k}, \\ & Z \geq 0, Z = Z^T, Z^2 = Z. \end{aligned} \tag{1}$$

We first cite a result from [20] without proof.

Proposition 2.1. *Finding a global solution of the classical K-means clustering equals to solving the 0-1 SDP problem (1) where $w_{ij} = x_i^T x_j$.*

A straightforward extension of the above model is to use other kernel matrices rather than the one in the classical K-means. For example, if we choose

$$w_{ij} = \phi(x_i, x_j) = \exp^{-\frac{\|x_i - x_j\|^2}{\sigma}}, \quad \sigma > 0, \tag{2}$$

then we end up with the so-called kernel K-means.

Except for the above mentioned cases, the 0-1 SDP model can also be applied to the so-called balanced clustering [3] where the number of points in every cluster is restricted. One special case of balanced clustering is requiring the number of points in every cluster must be equal to or large than a prescribed quantity $\tilde{n}$. It has been observed in [20] that such a requirement can be incorporated into the above 0-1 SDP by adding extra constraints $Z_{ii} \leq \frac{1}{\tilde{n}}$ to (1), which leads to the following problem

$$\begin{aligned} \min \quad & \mathrm{Tr(W(I-Z))} \\ & Z_{ii} \leq \frac{1}{\tilde{n}}, \quad i = 1, \cdots, n, \\ & Ze = e, \mathrm{Tr(Z) = k}, \\ & Z \geq 0, Z^2 = Z, Z = Z^T. \end{aligned} \tag{3}$$

2.3 0-1 SDP Model for K-ways Normalized Cut

Recently, the k-ways normalized cut received much attention in the machine learning community, and many interesting results about such an approaches have been reported [5, 9, 17, 18, 21, 25, 26]. In particular, Dhillon [5] at'al showed that the normalized cut is equivalent to the weighted kernel K-means. Xing and Jordan [26] considered an SDP relaxation for normalized k-cut. We next show that the k-ways normalized cut can be embedded into the 0-1 SDP model. Let us first recall the exact model for normalized k-cut [26]. Let W be the affinity matrix defined by (2) and $X = [x_{ij}] \in \Re^{n \times k}$ be the assignment matrix defined by

$$x_{ij} = \begin{cases} 1 & \text{If } x_i \text{ is assigned to } C_j; \\ 0 & \text{Otherwise,} \end{cases}$$

e be the all 1 vector in suitable space and $\mathcal{F}_k$ be a set defined by

$$\mathcal{F}_k = \{X : Xe^k = e^n, x_{ij} \in \{0, 1\}\}.$$

Let $d = We^n$ and $D = \mathrm{diag}\,(d)$. The exact model for normalized k-cut in [26] can be rewritten as

$$\min_{X \in \mathcal{F}_k} \quad \mathrm{Tr}\big(\mathrm{D^{-1}W - (X^TDX)^{-1}X^TWX}\big) \tag{4}$$

If we define

$$Z = D^{\frac{1}{2}} X (X^T D X)^{-1} X^T D^{\frac{1}{2}}, \tag{5}$$

then we have

$$Z^2 = Z, Z^T = Z, Z \geq 0, Z d^{\frac{1}{2}} = d^{\frac{1}{2}}.$$

Therefore, we can rewrite the objective function in (4) as a linear function via using matrix argument. By adding the above conditions for the matrix Z, we obtain the following 0-1 SDP problem:

$$\begin{aligned} \min \quad & \mathrm{Tr}\Big(\mathrm{D^{-\frac{1}{2}}WD^{-\frac{1}{2}}(I - Z)}\Big) && (6) \\ & Zd^{\frac{1}{2}} = d^{\frac{1}{2}}, \mathrm{Tr(Z) = k}, && (7) \\ & Z \geq 0, Z^2 = Z, Z = Z^T. && (8) \end{aligned}$$

We have

Theorem 2.2. *Finding a global solution of problem (4) equals to solving the 0-1 SDP problem (6).*

Proof. To prove the theorem, we first note that any feasible solution for problem (4) can be transferred into as a feasible solution of problem (6). If remains to show that from any feasible solution of problem (6), we can construct a feasible assignment matrix X for problem (4).

Suppose that Z is a feasible solution of problem (6). We can define a matrix $\bar{Z} = D^{-\frac{1}{2}} Z D^{\frac{1}{2}}$. It follows from the constraints (7)-(8) that

$$\bar{Z}^2 = \bar{Z}, \bar{Z}e = e, \bar{Z} \geq 0. \tag{9}$$

Let

$$\bar{z}_{i_0 j_0} = \arg\max \bar{z}_{ij}, \quad \mathcal{J}_0 = \{i : \bar{z}_{ij_0} > 0\}.$$

Since $\bar{Z}^2 = \bar{Z}$ and $\sum_{j=1}^n \bar{z}_{i_0 j} = 1$, it must hold

$$\begin{aligned} \bar{z}_{ij_0} \quad & = \bar{z}_{i_0 j_0}, \quad \forall i \in \mathcal{J}_0, \\ \sum_{j \in \mathcal{J}_0} \bar{z}_{i_0 j} \quad & = 1, \quad \bar{z}_{i_0 j} = 0 \quad \forall j \notin \mathcal{J}_0. \end{aligned}$$

Recall the definition of the matrix $\bar{Z}$, we can decompose the matrix $\bar{Z}$ into a bock matrix with the following structure

$$Z = \left(\begin{array}{cc} \bar{Z}_{\mathcal{J}_0\mathcal{J}_0} & 0 \\ 0 & \bar{Z}_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0} \end{array} \right), \tag{10}$$

where $\bar{\mathcal{J}}_0 = \{i : i \notin \mathcal{J}_0\}$. Now we claim that $\bar{Z}_{\mathcal{J}_0\mathcal{J}_0}$ is a submatrix with rank 1 for which all the elements in any column are equivalent. To see this, let us choose any column from the submatrix $\bar{Z}_{\mathcal{J}_0\mathcal{J}_0}$ and consider the minimum element in that column, i.e., for a fixed $j \in \mathcal{J}_0$,

$$\bar{z}_{i_1 j} = \arg\min_{i \in \mathcal{J}_0} \bar{z}_{ij}.$$

From the relation (9) we have

$$\bar{z}_{i_1 j} = \sum_{k \in \mathcal{J}_0} \bar{z}_{i_1 k} \bar{z}_{kj} \geq \bar{z}_{i_1 j} \sum_{k \in \mathcal{J}_0} \bar{z}_{i_1 k} = \bar{z}_{i_1 j},$$

and the equality holds if and only if all the element in the column are equivalent. Since $\bar{Z}_{\mathcal{J}_0\mathcal{J}_0}$ is a rank one matrix and the sum of each row equals 1, its trace also equals 1.

From the above discussion we can see that we can put all the points associated with the index set $\mathcal{J}_0$ into one cluster, and reduce the corresponding 0-1 SDP model (6) to a smaller problem as follows

$$\begin{array}{ll} \min & \mathrm{Tr}\Big(\mathrm{D}_{\bar{\mathcal{J}}_0}^{-\frac{1}{2}} \mathrm{W}_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0} \mathrm{D}_{\bar{\mathcal{J}}_0}^{-\frac{1}{2}} (\mathrm{I} - \mathrm{Z}_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0})\Big) \\ & Z_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0} d_{\bar{\mathcal{J}}_0}^{\frac{1}{2}} = d_{\bar{\mathcal{J}}_0}^{\frac{1}{2}}, \mathrm{Tr}\big(\mathrm{Z}_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0}\big) = \mathrm{k} - 1, \\ & Z_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0} \geq 0, Z^2_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0} = Z_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0}, Z_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0} = Z^T_{\bar{\mathcal{J}}_0\bar{\mathcal{J}}_0}. \end{array}$$

Repeating the above process, we can reconstruct all the clusters from a solution of problem (6). This establishes the equivalence between the two models (4) and (6). □

It should be noted that the only difference between (1) and (6) is the introduction of the scaling vector d. This coincides with the observation in [5]. However, we would like to point out that it is not so easy to add extra constraints such as the number of points in each cluster to model (6), while adding balanced constraint to (1) is a trivial task.

3 Approximate Algorithms for Solving 0-1 SDP

In this section we discuss how to solve the 0-1 SDP model for spectral clustering. For simplification of our discussion, we restrict us to the following unified model

$$\begin{aligned} \min \quad & \mathrm{Tr(W(I-Z))} && (11)\\ & Zs = s, \mathrm{Tr(Z) = k}, && (12)\\ & Z \geq 0, Z^2 = Z, Z = Z^T, && (13) \end{aligned}$$

where s is a positive scalar vector satisfying $\|s\| = 1$. Throughout the paper, we further assume that the underlying matrix W is positive semidefinite. This assumption is satisfied in both the MSSC model (1) and the normalized k-cuts (6).

The section consists of two parts. In the first subsection, we give a general introduction to algorithms for solving (11) and then describe an approximation algorithm based on PCA. In the second part, we propose a new approximation method for (11) based on projected PCA.

3.1 Approximation Algorithms for 0-1 SDP Based on PCA

We first describe a general scheme of approximation algorithms for (11).

Approximation Algorithm Based on Relaxation

Step 1: Choose a relaxation model for (11),

Step 2: Solve the relaxed problem for an approximate solution,

Step 3: Use a rounding procedure to extract a feasible solution to (11) from the approximate solution.

The relaxation step has an important role in the whole algorithm. For example, if the approximation solution obtained from Step 2 is feasible for (11), then it is exactly an optimal solution of (11). On the other hand, when the approximation solution is not feasible regarding (11), we have to use a rounding procedure to extract a feasible solution.

Various relaxations and rounding procedures have been proposed for solving (11) in the literature. For example, in [19], Peng and Xia considered a relaxation of (11) based on linear programming and a rounding procedure was also proposed in that work. Xing and Jordan [26] considered the SDP relaxation for normalized k-cuts and proposed a rounding procedure based on the singular value decomposition of the solution Z of the relaxed problem,i.e., $Z = U^T U$. In their approach, every row of U^T is cast as a point in the new space, and then the weighted K-means clustering is performed over the new data set in $\Re^k$. Similar works for spectral clustering can also be found in [9, 17, 18, 25, 27] where the singular value decomposition of the underlying matrix W is used and a K-means-type clustering based on the eigenvectors of W is performed. In the above-mentioned works, the solutions obtained from the weighted K-means algorithm for the original problem and that based on the eigenvectors of W has been compared, and simple theoretical bounds have been derived based on the eigenvalues of W.

The idea of using the singular value decomposition of the underlying matrix W is natural in the so-called principal component analysis (PCA) [13]. In [4], the link between PCA and K-means clustering was also explored and simple bounds were derived. In particular, Drineas et'al [6] proposed to use singular value decomposition to form a subspace, and then perform K-means clustering in the subspace $\Re^k$. They proved that the solution obtained by solving the K-means clustering in the reduced space can provide a 2-approximation to the solution of the original K-means clustering.

In what follows we consider an approximation algorithm similar to what reported in [26], which is based on the SDP relaxation of (11). There are several different ways to relax the 0-1 SDP model (11). First of all, the constraint $Z^2 = Z$ implies that Z must be a projection matrix. This implies $0 \preceq Z \preceq I$. Since we further require $Z \geq 0$, which indicates that there exists exactly one nonnegative eigenvector corresponding to the largest eigenvalue of Z (see Theorem 1.3.2 of [2]). On the other hand, since s is a nonnegative eigenvector of Z. It follows immediately that the largest eigenvalue of Z equals 1. In such a case, the constraint $Z \preceq I$ becomes superfluous and can be waived without any influence. Therefore, we need only to consider the following relaxed SDP problem

$$\begin{aligned} \min \quad & \mathrm{Tr(W(I - Z))} \\ & \mathrm{Tr(Z) = k, Z \succeq 0,} \\ & Zs = s, Z \geq 0. \end{aligned} \tag{14}$$

The above problem is feasible and bounded below. We can apply many existing optimization solvers such as interior-point methods to solve (14). However, we would like to point out here that although there exist theoretically polynomial algorithms for solving (14), most of the present optimization solvers are unable to handle the problem in large size efficiently.

By further removing the nonnegative requirement and the scaling constraint $Zs = s$ in (14), we obtain the following simple optimization problem

$$\begin{aligned} \min \quad & \mathrm{Tr(W(I - Z))} && (15) \\ & \mathrm{Tr(Z) = k, Z \succeq 0}, && (16) \end{aligned}$$

which can be solved by using the singular value decomposition of W. Denote

$$W = U\mathrm{diag}\,(\lambda_1, \cdots, \lambda_n)U^T,$$

where λ_i are the eigenvalues of W listed in the decreasing order, and U is an orthogonal matrix whose i-column is the eigenvector corresponding to λ_i. It follows immediately [26]

Theorem 3.1. *Suppose Z^* is a global solution to problem (11), we have*

$$\mathrm{Tr(W(I - Z*)) \geq Tr(W) - \sum_{i=1}^{k} \lambda_i}.$$

To extract a feasible solution to the original model, we use the k eigenvectors multiplied by the squared root of their corresponding eigenvalues, i.e., $\lambda_1^{\frac{1}{2}}u_1, \cdots, \lambda_k^{\frac{1}{2}}u_k$. Therefore, we get a matrix in $\Re^{n\times k}$ for which every row represent a point in $\Re^k$. We then perform the clustering task based on the new data set. From our analysis in Section 2, we know that this equals to solve the model (11) where the matrix W is replaced by its projection (W_k) onto the subspace generated by the eigenvectors $u_1, \cdots, u_k$, i.e.,

$$W_k = \sum_{i=1}^{k} \lambda_i u_i u_i^T.$$

The algorithm scheme can be described as follows:

Algorithm 1: Approximation Method based on PCA

S.1 Perform a singular value decomposition for the coefficient matrix W.

S.2 Calculate the matrix W_k by using the first k largest eigenvalues of W and their corresponding eigenvectors.

S.3 Solve the 0-1 SDP model with the coefficient matrix W_k and cluster the data set based on the obtained assignment matrix.

We have

Theorem 3.2. *Suppose Z^* is a global solution to the original problem (11) and Z_k^* is a global solution to the reduced problem (11) where W is replaced by W_k. Then we have*

$$\mathrm{Tr(W(I - Z_k^*)) \leq 2Tr(W(I - Z^*))}.$$

Proof. Let us define

$$U_k = \sum_{i=1}^{k} u_i u_i^T.$$

From Theorem 3.1, we have

$$\mathrm{Tr(W(I - U_k)) \leq Tr(W(I - Z^*))}.$$

Since

$$\mathrm{Tr(W(I - Z_k^*)) = Tr(W(I - U_k)) + Tr(W(U_k - Z_k^*))},$$

to prove the theorem, it suffices to show that

$$\mathrm{Tr(W(U_k - Z_k^*)) \leq Tr(W(I - Z^*))},$$

which can be stated as

$$\mathrm{Tr(W(I - U_k + Z_k^* - Z^*)) \geq 0}. \qquad (17)$$

From the choice of U_k, we have

$$\begin{aligned} W_k = WU_k = U_k W U_k, &\quad U_k = U_k^2; \\ W - W_k = W(I - U_k) = (I - U_k)W(I - U_k) &\quad (I - U_k)^2 = I - U_k. \end{aligned}$$

Since Z_k^* is a solution of problem (11) with a coefficient matrix W_k, we have

$$\mathrm{Tr(W_k(I - Z_k^*)) \leq Tr(W_k(I - Z^*))}. \qquad (18)$$

It follows

$$\begin{aligned} \mathrm{Tr(W(I - U_k + Z_k^* - Z^*))} &= \mathrm{Tr(WU_k(I - U_k + Z_k^* - Z^*))} \\ &\quad \mathrm{+Tr(W(I - U_k)(I - U_k + Z_k^* - Z^*))} \\ &= \mathrm{Tr(W_k(I - U_k + Z_k^* - Z^*))} \\ &\quad \mathrm{+Tr(W(I - U_k)(I + Z_k^* - Z^*))} \\ &\geq \mathrm{Tr(W_k(Z_k^* - Z^*)) + Tr(W(I - U_k)(I - Z^*))} \\ &\geq \mathrm{Tr((I - U_k)W(I - U_k)(I - Z^*)) \geq 0}, \end{aligned}$$

where the first inequality is given by the fact that $I - U_k \succeq 0$, the second inequality by (18) and the fact that $Z_k^* \succeq 0$, and the last one by $I - Z^* \succeq 0$. This proves the relation (17), which further yields the theorem. $\square$

We point out although we have proved that the algorithm based on PCA can provide a 2-approximation solution, it indeed requires to find the global solution of the reduced problem. In general, this is still a nontrivial task. For example, for the classical K-means clustering with a data set in $\Re^k$, Drineas et'al [6] proposed an algorithm that runs in $O(n^{k^3/2})$ time. Even for the special case $k = 2$, such an algorithm is clearly impractical for large data set.

3.2 An approximation Algorithm Based on Projected PCA

In this subsection, we propose a new approximation algorithm based another relaxation form of the model (11). Let us recall that in the relaxed model (14), we stipulate that s is an eigenvector of the final solution matrix Z. Since we already know this fact in advance, we can keep such a simple constraint in our relaxed problem. We therefore obtain another form of relaxation For example, if we remove only the nonnegative requirement in the relaxation form (14), then we obtain the following 0-1 SDP problem:

$$\begin{aligned} \min \quad & \mathrm{Tr(W(I-Z))} \\ & Zs = s, \mathrm{Tr(Z) = k}, \\ & Z \succeq 0. \end{aligned} \tag{19}$$

We next discuss how to solve the above problem. First, we note that for any feasible solution of (19), let us define

$$\bar{Z} = Z - ss^T.$$

It is easy to see that

$$\bar{Z} = (I - ss^T)Z = (I - ss^T)Z(I - ss^T), \tag{20}$$

i.e., $\bar{Z}$ represents the projection of the matrix Z onto the null subspace of s. Moreover, since $\|s\| = 1$, it is easy to verify that

$$\mathrm{Tr\big(\bar{Z}\big) = Tr(Z) - 1 = k - 1.}$$

Let $\overline{W}$ denote the projection of the matrix W onto the null space of s, i.e.,

$$\overline{W} = (I - ss^T)W(I - ss^T). \tag{21}$$

Then, we can reduce (19) to

$$\begin{aligned} \min \quad & \mathrm{Tr}\big(\overline{\mathrm{W}}(\mathrm{I}-\bar{\mathrm{Z}})\big) \\ & \mathrm{Tr}\big(\bar{\mathrm{Z}}\big) = \mathrm{k}-1, \\ & \bar{Z} \succeq 0. \end{aligned} \tag{22}$$

Let $\lambda_1, \cdots, \lambda_{n-1}$ be the eigenvalues of the matrix $\overline{W}$ listed in the order of decreasing values. The optimal solution of (22) can be achieved if and only if

$$\mathrm{Tr}\big(\overline{\mathrm{W}}\bar{\mathrm{Z}}\big) = \sum_{\mathrm{i}=1}^{\mathrm{k}-1} \lambda_{\mathrm{i}}.$$

This gives us an easy way to solve (22) and correspondingly (19). We call it projected PCA to differentiate it from the standard PCA. The algorithmic scheme for solving (19) can be described as follows:

Projected PCA

Step 1: Calculate the projection $\overline{W}$ via (21);

Step 2: Use singular value decomposition method to compute the first $k-1$ largest eigenvalues of the matrix $\overline{W}$ and their corresponding eigenvectors $u_1, \cdots, u_{k-1}$,

Step 3: Set

$$Z = ss^T + \sum_{i=1}^{k-1} u_i u_i{}^T.$$

From our above discussion, we immediately have

Theorem 3.3. *Let Z^* be the global optimal solution of (11), and $\lambda_1, \cdots, \lambda_{k-1}$ be the first $k-1$ largest eigenvalues of the matrix $\overline{W}$. Then we have*

$$\mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}^*)) \geq \mathrm{Tr}(\mathrm{W}) - \mathrm{s}^{\mathrm{T}}\mathrm{W}\mathrm{s} - \sum_{\mathrm{i}=1}^{\mathrm{k}-1} \lambda_{\mathrm{i}}.$$

In the sequel we propose a rounding procedure to extract a feasible solution for (11) from a solution of the relaxed problem (19) provided by the projected PCA. Our rounding procedure follows a similar vein as the rounding procedure in the previous subsection. In case no confusion occurs, we use the notation introduce in the previous subsection. Let

$$U_{k-1} = \sum_{i=1}^{k-1} u_i u_i^T$$

be the solution matrix obtained from the projected PCA, and

$$U_k = ss^T + U_{k-1}, \quad \overline{W}_{k-1} = \overline{W}(I - U_{k-1}).$$

We can formulate a matrix in $\Re^{n\times(k-1)}$ whose i-th column is $\lambda_i^{\frac{1}{2}} u_i$. Then we cast each row in such a matrix as a point in $\Re^{k-1}$. We thus obtain a data set of n points in $\Re^{k-1}$. Then we perform the clustering task for the new data set. In other words, we need to solve the 0-1 SDP model (11) with a new coefficient matrix $\overline{W}_{k-1}$. Finally, we partition all the points in the original space based on the obtained clusters for the new data set.

The whole algorithm can be described as follows.

Algorithm 2: Approximation Method based on Projected PCA

Step 1: Calculate the projected matrix $\overline{W}$ of the matrix W onto the null space of s;

Step 2: Use singular value decomposition to compute the first $k-1$ largest eigenvalues of the matrix $\overline{W}$ and their corresponding eigenvectors $u_1, \cdots, u_{k-1}$, and compute the matrix $\overline{W}_{k-1}$;

Step 3: Solve problem (11) with the coefficient matrix $\overline{W}_{k-1}$ and assign all the points in the original space based on the obtained assignment.

The above algorithm can be viewed as an improved version of the algorithm based on PCA. In particular, in case of bi-clustering, the subproblem involved in the above algorithm needs only to cluster a data set in one dimension, which can be done in $O(n \log n)$ time. For detailed discussion, we refer the readers to the refined K-means in one dimension described in [20]. This improves the efficiency of the algorithm substantially and allows us to deal with large-scale data set.

We also point out that such an idea has been employed by Shi and Malik [21] in their seminar paper on normalized cut for image segmentation. In that case, the 0-1 SDP model takes the form as in (6) with $k = 2$. Since $d^{\frac{1}{2}}$ is the eigenvector corresponding to the largest eigenvalue of the underlying coefficient matrix, Shi and Malik proposed to use the eigenvector corresponding to second largest eigenvalue of the coefficient matrix to cluster the data set.

If $k \geq 3$, then the subproblem in Algorithm 2 is still nontrivial. For the classical K-means clustering, we can resort to the algorithm in [6] to solve problem (11) in low dimensional space. It is easy to see that the algorithm takes $O(n^{k^2(k-1)/2})$ time to find the global solution of the subproblem in Step 3 of Algorithm 2, which is roughly a $\frac{1}{n^{k^2/2}}$ fraction of the running time when the same procedure is applied to solve the subproblem in [6]. This is because the working space in our algorithm is one dimension less than the space in [6].

We next progress to estimate the solution obtained from Algorithm 2. We have

Theorem 3.4. *Suppose that Z^* is a global solution to problem (11) and Z_k^* is the solution provided by Algorithm 2. Then, we have*

$$\mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}_\mathrm{k}^*)) \le 2\mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}^*)).$$

Proof. Let Z^* be a global solution to (11) and Z_k^* is the solution provided by Algorithm 2. From the choices of U_{k-1} and U_k it follows

$$\begin{aligned} \mathrm{Tr}\big((\mathrm{I}-\mathrm{U_k})\overline{\mathrm{W}}_{\mathrm{k-1}}\big) &= 0; & (23)\\ \mathrm{Tr}\big(\mathrm{U_k}(\overline{\mathrm{W}}-\overline{\mathrm{W}}_{\mathrm{k-1}})\big) &= 0.. & (24)\end{aligned}$$

From Theorem 3.3, we have

$$\mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}^*)) \ge \mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{U_k})). \tag{25}$$

It follows

$$\mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}_\mathrm{k}^*)) = \mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{U_k}+\mathrm{U_k}-\mathrm{Z}_\mathrm{k}^*)) \le \mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}^*))+\mathrm{Tr}(\mathrm{W}(\mathrm{U_k}-\mathrm{Z}_\mathrm{k}^*)).$$

To prove the conclusion in the theorem, it suffices to show

$$\mathrm{Tr}(\mathrm{W}(\mathrm{U_k}-\mathrm{Z}_\mathrm{k}^*)) \le \mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}^*)), \tag{26}$$

or equivalently

$$\mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}^*+\mathrm{Z}_\mathrm{k}^*-\mathrm{U})) \ge 0. \tag{27}$$

By the choices of Z^*, Z_k^* and U_k, it is easy to verify

$$(I-Z^*+Z_k^*-U)s = 0, \tag{28}$$
$$(I-ss^T)(I-Z^*+Z_k^*-U)(I-ss^T) = I-Z^*+Z_k^*-U). \tag{29}$$

It follows immediately that

$$\begin{aligned}
\mathrm{Tr}(\mathrm{W}(\mathrm{I}-\mathrm{Z}^*+\mathrm{Z}_\mathrm{k}^*-\mathrm{U})) &= \mathrm{Tr}\big(\overline{\mathrm{W}}(\mathrm{I}-\mathrm{Z}^*+\mathrm{Z}_\mathrm{k}^*-\mathrm{U}_\mathrm{k})\big)\\
&= \mathrm{Tr}\big(\overline{\mathrm{W}}_{\mathrm{k}-1}(\mathrm{I}-\mathrm{Z}^*+\mathrm{Z}_\mathrm{k}^*-\mathrm{U}_\mathrm{k})\big)\\
&\quad +\mathrm{Tr}\big((\mathrm{I}-\mathrm{Z}^*+\mathrm{Z}_\mathrm{k}^*-\mathrm{U}_\mathrm{k})(\overline{\mathrm{W}}-\overline{\mathrm{W}}_{\mathrm{k}-1})\big)\\
&= \mathrm{Tr}\big((\mathrm{Z}_\mathrm{k}^*-\mathrm{Z}^*)\overline{\mathrm{W}}_{\mathrm{k}-1}\big)+\mathrm{Tr}\big((\mathrm{I}-\mathrm{Z}^*+\mathrm{Z}_\mathrm{k}^*)(\overline{\mathrm{W}}-\overline{\mathrm{W}}_{\mathrm{k}-1})\big)\\
&\geq \mathrm{Tr}\big(\overline{\mathrm{W}}_{\mathrm{k}-1}(\mathrm{Z}_\mathrm{k}^*-\mathrm{Z}^*)\big),
\end{aligned}$$

where the last equality is given by (23) and (24), and the last inequality is implied by the fact that $I-Z^*+Z_k^*\succeq 0, \overline{W}-\overline{W}_{k-1}\succeq 0$. Recall that Z_k^* is the global solution of problem (11) with the coefficient matrix $\overline{W}_{k-1}$ while Z^* is only a feasible solution of the same problem, we therefore have

$$\mathrm{Tr}\big(\overline{\mathrm{W}}_{\mathrm{k}-1}(\mathrm{Z}_\mathrm{k}^*-\mathrm{Z}^*)\big)\geq 0,$$

which further implies (26). This finishes the proof of the theorem. □

It should be mentioned that our above results can be extended to scenario of constrained clustering where the number of points in every cluster is bounded. In such a case, we need to add some extra constraints on the size of the clusters in the subproblem involved in Algorithm 2. Since the discussion for constrained clustering follows a similar chain of reasoning as in the proofs of Theorem 3.4, we leave the details to interested readers.

Before we close this section, we discuss briefly the complexity of Algorithm 2. In the first step of the algorithm, we need to perform the singular value decomposition for the matrix $\overline{W}$. In general, this takes $O(n^3)$ time. If we use the power method [8] to calculate the first $k-1$ largest eigenvalues and their corresponding eigenvectors, then the complexity can be reduced to $O(kn^2)$. However, in the context of the classical K-means clustering, we can use the structure of the underlying matrix W to speed up the process. Recall that for K-means clustering, we have $W=W_xW_x^T$ where $W_x\in\Re^{n\times m}$ is a matrix such that every row represents a point in $\Re^m$. In such a case, it is not necessary to calculate the matrix W to estimate its eigenvalues and eigenvectors exactly. Note that we can perform a singular value decomposition on the matrix W_x directly, which can proceed in the following way. We first compute a matrix $\tilde{W}=W_x^TW_x\in\Re^{m\times m}$ which takes $O(nm^2)$ time. It is easy to see that the matrix $\tilde{W}$ has the same spectrum as that of the matrix W. Therefore, we can perform a singular value decomposition on $\tilde{W}$ directly such that

$$\tilde{W}=V\operatorname{diag}\lambda_1,\cdots,\lambda_m V^T,$$

where $V \in \Re^{m\times m}$ is an orthogonal matrix such that every column is an eigenvector of $\tilde{W}$. This takes $O(m^3)$ time. We then calculate the matrix $\tilde{U} = W_xV$ in $O(nm^2)$ time. One can easily verify that the i-th column of the matrix $\tilde{U}$ is an eigenvector of W corresponding to eigenvalue λ_i. Therefore, the total computational cost to calculate the eigenvalues and their corresponding eigenvectors is $O(nm^2 + m^3)$. If m is not very large, say $m < 1000$ (which is true for most data sets in practice), then we can obtain the eigenvalues and eigenvectors of W very quickly. However, this is not true if we use some other kernel matrices such as in normalized cut.

Next we discuss the complexity in solving the subproblem in Algorithm 2. We can use the hierarchical as suggested in [21] and perform a bi-clustering task subsequently. As described in [20], the subproblem in Algorithm 2 for bi-clustering can be done in $O(n \log n)$ time. Therefore, the total complexity of the algorithm will be $O(n \log n + nm^2 + m^3)$. This allows us to cope with extremely large data set in a reasonably high dimension ($m < 1000$).

4 Conclusions

In the present work, we presented a novel unified framework for spectral clustering and proposed two different approximation algorithms for solving the unified 0-1 SDP model based on PCA and projected PCA, respectively. We have shown that although both algorithms can provide a 2-approximation to the original clustering problem, the algorithm based on projected PCA is more efficiently. Our results not only open new avenues for solving spectral clustering, but also provide insightful analysis for several existing algorithms in the literature.

There are several open questions regarding the new 0-1 SDP model. First, we note that there are several different ways to relax the 0-1 SDP model that have not been investigated. For example, we can solve the relaxed model (14) to find an approximation solution, which will give us a tighter bound than the relaxation based on PCA and projected PCA. However, it is unclear how to design a rounding procedure to extract a feasible clustering and how to estimate the quality of the extracted solution. Secondly, both algorithms in the present paper require to solve the subproblems exactly, which turns out still to be a challenge. More study is necessary to address these questions.

References

[1] Bach, F.R. and Jordan, M.I. Learning spectral clustering, *Advances in Neural Information Processing Systems (NIPS)*, 16, 2004.

[2] Berman, A. and Plemmins, R.J. (1994). *Nonnegative matrices in the mathematical sciences*, SIAM Classics in Applied Mathematics, Philadelphia.

[3] Bradley, P., Bennet, K. and Demiriz, A.,(2000) Constrained K-Means Clustering. *MSR-TR-2000-65*, Microsoft Research.

[4] Ding, C. and He, X. (2004). K-means clustering via principal component analysis. *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada.

[5] Dhillon, I.S., Guan, Y. and Kulis, B. Kernel k-means, Spectral Clustering and Normalized Cuts. *Proceedings of The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD)*, 551-556, August 2004.

[6] Drineas, P., Frieze, A., Kannan, R., Vempala, R. and Vinay, V. (2004). Clustering large graphs via singular value decomposition. *Machine Learning*, 56, 9-33.

[7] Ghosh J.(2003). Scalable Clustering. In N. Ye, Editor, The Handbook of Data Mining, Lawrence Erlbaum Associate, Inc, pp. 247-277.

[8] Gulub, G. and Loan, C. V. (1996) *Matix Computation*. John Hopkins University Press.

[9] Gu, M., Zha, H., Ding, C., He, X. and Simon, H. (2001). Spectral relaxation models and structure analysis for k-way graph Clustering and bi-clustering. Penn State Univ Tech Report.

[10] Hansen, P., Jaumard, B. and Mladenović, N. (1998). Minumum sum of squares clustering in a low dimensional space. *J. Classification*, 15, 37-55.

[11] Jain, A.K., & Dubes, R.C. (1988). *Algorithms for clustering data*. Englewood Cliffs, NJ: Prentice Hall.

[12] Jain, A.K., Murty, M.N. and Flynn, P.J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31, 264-323.

[13] Jolliffe, I. (2002). *Principal component analysis.* Springer, 2nd edition.

[14] Kaufman, L. and Peter Rousseeuw, P. (1990). Finding Groups in Data, an Introduction to Cluster Analysis, John Wiley.

[15] Karisch, S.E. and Rendl, F. (1998). Semidefinite programming and graph equipartition. *Fields Institute Communications.* 18, 77-95.

[16] McQueen, J.(1967). Some methods for classification and analysis of multivariate observations. *Computer and Chemistry*, 4, 257-272.

[17] Meila, M. and Shi, J. (2001). A random walks view of spectral segmentation. Int'l Workshop on AI & Stat.

[18] Ng, A.Y., Jordan, M.I. and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Proc. Neural Info. Processing Systems, NIPS*, 14.

[19] Peng, J.M.. and Xia, Y. A new theoretical framework for K-means clustering, To appear in *Foundation and recent advances in data mining*, Eds Chu and Lin, Springer Verlag, 2005.

[20] Peng, J. and Wei, Y. (2005). Approximating K-means-type clustering via semidefinite programming, Technical Report, Department of CAS, McMaster University, Ontario, Canada.

[21] Shi,J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 22, 888-905.

[22] Späth, H. (1980). *Algorithms for Data Reduction and Classification of Objects*, John Wiley & Sons, Ellis Horwood Ltd.

[23] Verma, D. and Meila, M. Comparison of spectral clustering methods, Technical Report, Department of Statistics, University of Washington, Seattle, 2005.

[24] Ward, J.H. (1963). Hierarchical grouping to optimize an objective function. *J. Amer. Statist. Assoc.,* 58, 236-244.

[25] Weiss, Y. (1999). Segmentation using eigenvectors: a unifying view. *Proceedings IEEE International Conference on Computer Vision*, 975-982.

[26] Xing, E.P. and Jordan, M.I. (2003). On semidefinite relaxation for normalized k-cut and connections to spectral clustering. Tech Report CSD-03-1265, UC Berkeley.

[27] Zha, H., Ding, C., Gu, M., He, X. and Simon, H. (2002). Spectral Relaxation for K-means Clustering. In Dietterich, T., Becker, S. and Ghahramani, Z. Eds., *Advances in Neural Information Processing Systems 14*, pp. 1057-1064. MIT Press.