Clustering for Bioinformatics via Matrix Optimization*

Suely Oliveira
Department of Computer Science
University of Iowa
Iowa City, Iowa 52242, USA
oliveira@cs.uiowa.edu

David E. Stewart
Department of Mathematics
University of Iowa
Iowa City, Iowa 52242, USA
dstewart@math.uiowa.edu

ABSTRACT

A new matrix-based clustering method is presented that is able to handle medium to large data sets that is related to semi-definite programming techniques. The method proposed involves solving a non-convex optimization problem. The problem of local minima that are far from global minima, however, does not appear to be a great difficulty. The method was applied to a well known biological clustering problem, and appears to produce consistent clusterings that are close to the original claimed clustering.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Bioinformatics

Keywords

clustering, optimization, semi-definite programming

1. INTRODUCTION

Clustering and classifying data items are common tasks in mining data sets [4, 12, 14, 1], both for text data mining and mining biological data sets. Clustering tasks are typically separated from classification tasks in that classification starts with a known collection of groups, and the task is to assign each data item into a group. This is often called "supervised clustering" because the group or cluster of certain data items is pre-specified. In this way, the assignment of data items to clusters is "supervised" by the user of the classification method. Clustering on the other hand does not have a pre-specified collection of groups or clusters with certain data items already assigned to a cluster. Instead, the task is to identify what groups should exist, and even the number of groups.

With this understanding, clustering is much more appropriate for exploratory data analysis, while classification is more appropriate where there is already considerable information available about the structure of the data. In truth, there is a spectrum of tasks between these two extremes depending on how many of the data items in the data set are pre-assigned. If most of the data items are pre-assigned to certain clusters, then determining the assignment of the remaining data items can be done by looking at the "neighboring" data items. This is the basis of the *k-nearest neighbors* (kNN) algorithm [6, 21]. However, if few of the data items are pre-assigned to clusters, then many data items will probably be far from these few exemplars, and the boundaries of the clusters must be determined from the structure of the collection of unassigned data items. This makes the classification task look more like an unsupervised clustering problem.

In this paper we focus on the unsupervised clustering problem. The methods described in this paper can be extended to the supervised classification problem; it is expected that these algorithms will perform well for classification problems where only a few data items are pre-assigned to a cluster.

The starting point for a clustering problem is typically a set of data points (or items) $\mathbf{x}_i \in \mathbb{R}^m$, $i = 1, 2, \ldots, N$. In certain problems we have the issue of *missing data* where each value of the data point may be a known number, or listed as missing. That is $\mathbf{x}_i \in (\mathbb{R} \cup \{\perp\})^m$ where "$\perp$" indicates a missing value. In this paper we will concentrate mainly on data sets with complete data. An alternative starting point for clustering is a collection of distance or dissimilarity values d_{ij} where both i and j range over the data items. Typically $d_{ij} = \phi\left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)$ for a suitable monotonic function ϕ; the most common choices are $\phi(s) = s$ and $\phi(s) = s^2$.

However the data is provided, clustering with an unspecified number of clusters is an inherently ill-defined task. Just as the classification of living things can be broken into a finer or coarser groups according to the level of refinement and detail desired, any cluster can be broken down into smaller parts. In doing so small differences can be used to do this, even though they are likely to be merely "noise". The question is a statistical one and can be addressed, at least partially, in terms of information theory. These issues will not be addressed in detail in this paper.

1.1 Clustering via optimization

A common approach to clustering is to represent the clustering problem as an optimization problem. This approach goes back to Rao [17]. The advantages of these methods is that they are fairly simple to understand: one formulates an objective function which represents the "quality" of the clustering, and then looks for algorithms to find the minimizer of this objective function. The main difficulty with this approach to clustering is that the problems are discrete optimization problems and are typically combinatorially hard [7].

For example, the k-means clustering approach for K clusters is to

*

minimize

$$\sum_{i=1}^{N} \left\| \mathbf{x}_i - \mathbf{c}_{a(i)} \right\|^2$$

over all all "centers" $\mathbf{c}_j \in \mathbb{R}^m$, $j = 1, 2, \ldots, K$, and assignments $a\colon \{1,2,\ldots,N\} \to \{1,2,\ldots,K\}$. The standard k-means *algorithm* (also known as Lloyd's algorithm [10]) is an iteration that first updates the centers $\mathbf{c}_j$ to be the mean of all $\mathbf{x}_i$ where $a(i) = j$, and then updates the assignment so that $a(i) = j$ where $\mathbf{c}_j$ is the closest center to $\mathbf{x}_j$. This algorithm amounts to an iteration where the minimizing centers $\mathbf{c}_j$ and assignments $a\colon \{1,2,\ldots,N\} \to \{1,2,\ldots,K\}$ are a fixed-point of the iteration. Alternative methods that have been explored include local, stochastic and evolutionary search algorithms such as assignment swapping, simulated annealing, genetic algorithms, and related heuristics [8, 25].

1.2 Semi-definite programs

Recently there have been a number of developments in the area of optimization that have the potential to revolutionize the approach to solving the combinatorially hard optimization problems that arise in clustering problems. These have centered around the use of matrices as the primary variables in certain structured optimization problems. The best known of these problems are semi-definite programming (SDP) problems [19, 23]. These problems have the form

$$\min_X C \bullet X \qquad \text{subject to} \tag{1.1}$$

$$A_i \bullet X = b_i, \qquad i = 1, 2, \ldots, m, \tag{1.2}$$

$$X \succeq 0, \tag{1.3}$$

where the minimum is taken over all X that are symmetric $n \times n$ matrices. The expression "$A \bullet B$" is the Frobenius inner product of the matrices A and B:

$$A \bullet B = \operatorname{trace}(A^T B) = \sum_{i,j} a_{ij} b_{ij}.$$

In general, the inequality "$A \succeq B$" means "$\mathbf{z}^T A \mathbf{z} \geq \mathbf{z}^T B \mathbf{z}$ for all $\mathbf{z}$". Equivalently, "$A \succeq B$" means that "$A - B$ is positive semi-definite".

Thus (1.1–1.3) is the problem of minimizing a linear function of X subject to linear constraints and that X is a positive semi-definite matrix; that is, $\mathbf{z}^T X \mathbf{z} \geq 0$ for all $\mathbf{z}$. Also, the set $\{X \mid X \succeq 0\}$ is a convex set: if $A, B \succeq 0$ then for $0 \leq \theta \leq 1$, $\theta A + (1-\theta) B \succeq 0$. Thus (1.1–1.3) is a convex optimization problem; that is, the objective function is convex, and the set of points satisfying the constraints is also convex.

Convex optimization problems are easier to work with since every local minimizer (or indeed, critical point) is a global minimizer. In particular, the problem of many irrelevant local minima does not occur with convex optimization problems. Semi-definite programs can be solved in essentially polynomial time by practical algorithms [2, 9, 19]. Semi-definite programs also provide a means of approximating hard combinatorial problems [13, 15, 22, 23, 24], including clustering problems ([16] for k-means).

2. AN OPTIMIZATION FORMULATION

An optimization formulation for clustering first proposed by Rao [17] is a matrix-based approach to clustering. The basic data for this approach is not the data set $\mathbf{x}_i$, $i = 1, 2, \ldots, N$, but rather distance or dissimilarity data d_{ij} representing the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. In Rao's formulation of the clustering problem the main variable is a $N \times K$ matrix Z; $z_{il} = 1$ means that data item i belongs to cluster l, and $z_{il} = 0$ otherwise. The task is to find the minimizer of

$$\sum_{i,j=1}^{N} d_{ij} \sum_{l=1}^{K} z_{il} z_{jl} \qquad \text{subject to} \tag{2.1}$$

$$\sum_{l=1}^{K} z_{il} = 1 \qquad \text{for all } i, \tag{2.2}$$

$$\sum_{i=1}^{N} z_{il} \geq 1 \qquad \text{for all } l, \tag{2.3}$$

$$z_{il} \in \{0, 1\} \qquad \text{for all } i, l. \tag{2.4}$$

Note that $\sum_{l=1}^{K} z_{il} z_{jl} = 1$ if data items i and j belong to the same cluster, and zero otherwise. The constraint $\sum_{l=1}^{K} z_{il} = 1$ means that data item i is assigned to exactly one cluster. The constraint $\sum_{i=1}^{N} z_{il} \geq 1$ says that each cluster has at least one data item. As given, this is a binary quadratic optimization problem. We assume that $d_{ii} = 0$; that is, the distance or dissimilarity between any object and itself is zero.

2.1 Relaxations and simplifications

Solving (2.1–2.4) directly is likely to be difficult. Instead we relax some of the constraints to obtain a simpler and easier to solve problem. First we remove the constraint $\sum_{i=1}^{N} z_{il} \geq 1$ that each cluster has at least one data item. This allows clusters with no elements. Essentially this allows any number of clusters up to K. If $K \geq N$, then the objective function can be set to zero by having N clusters each containing exactly one data item.

The most important constraints are that $z_{il} \in \{0, 1\}$ for all i, l, and that $\sum_{l=1}^{K} z_{il} = 1$ for all i. If we convexify these constraints we get $z_{il} \geq 0$ for all i, l and $\sum_{l=1}^{K} z_{il} = 1$ for all i. This is equivalent to requiring that each row of Z belongs to a unit simplex.

The objective function can be conveniently written in terms of standard matrix operations: $\sum_{l=1}^{K} z_{il} z_{jl} = \left(ZZ^T\right)_{ij}$, so the objective function is

$$\begin{aligned} \sum_{i,j=1}^{N} d_{ij} \sum_{l=1}^{K} z_{il} z_{jl} &= D \bullet \left(ZZ^T\right) \\ &= \operatorname{trace}\left(Z^T D Z\right). \end{aligned}$$

This *not* a convex function unless D is a positive semi-definite matrix. In clustering problems, since $d_{ii} = 0$ and $d_{ij} > 0$ for $i \neq j$, then D is not a positive semi-definite matrix.

These relaxations of the constraints lead to the following optimization problem:

$$\min_Z D \bullet \left(ZZ^T\right) \qquad \text{subject to} \tag{2.5}$$

$$Z\mathbf{e}^{(K)} = \mathbf{e}^{(N)}, \tag{2.6}$$

$$Z \geq 0 \qquad \text{(componentwise)}, \tag{2.7}$$

where $\mathbf{e}^{(m)}$ is the m-dimensional vector of ones. Since we obtain this optimization problem by relaxing (or removing) constaints of the original problem (2.1–2.4), we say that (2.5–2.7) is a relaxation of (2.1–2.4).

The optimization problem (2.5–2.7) is still not a convex optimization problem since $Z \mapsto D \bullet \left(ZZ^T\right)$ is not a convex function. We can make the objective function convex by writing it as $D \bullet Y$ with

$Y = ZZ^T$, since linear functions are convex. The problem with this is that the constraints $Y = ZZ^T$ are not linear, so that again the problem is not convex. However, if we carry out a further relaxation of "$Y = ZZ^T$", we can obtain a convex relaxation: "$Y \succeq ZZ^T$" is equivalent to the condition that

$$\begin{bmatrix} Y & Z \\ Z^T & I \end{bmatrix} \succeq 0. \tag{2.8}$$

Since the set of positive semi-definite matrices is a convex set, the set of pairs (Y,Z) satisfying (2.8) is a convex set. Combining the relaxations gives the following SDP:

$$\min_{Z,Y} D \bullet Y \qquad \text{subject to} \tag{2.9}$$

$$\begin{bmatrix} Y & Z \\ Z^T & I \end{bmatrix} \succeq 0, \tag{2.10}$$

$$Z\mathbf{e}^{(K)} = \mathbf{e}^{(N)}, \tag{2.11}$$

$$Z \geq 0. \tag{2.12}$$

It is also possible to add the constraint

$$Y \geq 0 \qquad \text{(componentwise)}. \tag{2.13}$$

Since the entries of D are distances, $D \geq 0$ componentwise; (2.13) then ensures that $D \bullet Y \geq 0$. A relaxation could also include the constraint $Y \geq ZZ^T$ componentwise, but this is not a convex constraint.

2.2 Symmetry properties

Both the original problem (2.1–2.4) and the relaxation (2.9–2.13) have a symmetry property: if P is a $K \times K$ permutation matrix and $Z' = ZP$, then

$$D \bullet \left(Z'Z'^T\right) = D \bullet \left(ZPP^TZ^T\right) = D \bullet \left(ZZ^T\right);$$

Y and Z' satisfies all the constraints if and only if Y and Z satisfy all the constraints: $Z \geq 0$ componentwise if and only if $ZP \geq 0$; $Z'\mathbf{e}^{(K)} = ZP\mathbf{e}^{(K)} = Z\mathbf{e}^{(K)} = \mathbf{e}^{(N)}$;

$$\begin{bmatrix} Y & Z' \\ Z'^T & I \end{bmatrix} = \begin{bmatrix} I & \\ & P^T \end{bmatrix} \begin{bmatrix} Y & Z \\ Z^T & I \end{bmatrix} \begin{bmatrix} I & \\ & P \end{bmatrix} \succeq 0.$$

Intuitively, this amounts to saying that permuting the cluster numbers does not change the problem. For supervised classification problems, this is no longer true, since some data items are pre-assigned a cluster or group number. However, if the number of pre-assigned data items is small, it would be approximately true even then.

Because the convexified problem (2.9–2.13) has this property, the solution set also has this symmetry. For convex optimization problems, the solution set is a convex set; for *strictly* convex optimization problems (where the objective function is strictly convex, satisfying the strict inequality $\phi(\theta\mathbf{x}+(1-\theta)\mathbf{y}) < \theta\phi(\mathbf{x})+(1-\theta)\phi(\mathbf{y})$) whenever $\mathbf{x} \neq \mathbf{y}$ and $0 < \theta < 1$) the solution is unique. Strictly convex functions include $Z \mapsto Z \bullet Z$, and multiplying a strictly convex function by a positive number gives a strictly convex function, so minimizing $D \bullet Y + \varepsilon\,(Z \bullet Z + Y \bullet Y)$ over Z and Y subject to the constraints (2.10–2.13) for any $\varepsilon > 0$ has a unique solution. As $(ZP) \bullet (ZP) = \text{trace}\left(P^TZ^TZP\right) = \text{trace}\left(Z^TZ\right) = Z \bullet Z$ is also invariant under the symmetry, the unique solution of the strictly convex approximation also has this symmetry. Interior point methods for convex optimization problems typically converge to the centroid of the solution set, which for (2.9–2.13) must also have this symmetry.

Thus any reasonable method for the convex relaxation (2.9–2.13) will converge to a unique point which satisfies the symmetry $Z = ZP$. Unfortunately, this means that each row $\mathbf{z}^T$ of Z satisfies the symmetry $\mathbf{z}^T = \mathbf{z}^TP$. Since P can be any permutation matrix, this indicates that each entry of $\mathbf{z}^T$ must be the same. Thus $\mathbf{z}^T = \mathbf{e}^{(K)T}/K$. Since this is true for each row of Z we have the optimum $Z = \mathbf{e}^{(N)}\mathbf{e}^{(K)T}/K$. Since Z was meant to be the main decision variable in this formulation, it is disappointing that the optimal Z gives no information about the optimal clustering.

2.3 A non-convex relaxation

Instead of the fully convex relaxation, we consider a partial relaxation (2.5–2.7). This partial relaxation is a non-convex optimization problem, and so can have multiple local minima. This partial relaxation is also invariant under the transformation $Z \mapsto ZP$ where P is a permutation matrix. Thus we expect at least $K!$ local minima, each local minimizer being a relabeling of the others. There may be other local minima, but hopefully not many.

Another non-convex but partially convexified approximate problem can be obtained by replacing the objective in (2.5) by

$$\min_Z D \bullet \left(ZZ^T\right) + \alpha \|Z\|_{nuc} \tag{2.14}$$

where

$$\begin{aligned} \|Z\|_{nuc} &= \max_{U,V \text{ orthogonal}} U^TZV \\ &= \sum_{i=1}^{\min(K,N)} \sigma_i(Z), \end{aligned}$$

with $\sigma_i(A)$ the ith singular value of A. The norm $\|Z\|_{nuc}$ is called the nuclear norm of Z. The addition of $\alpha\|Z\|_{nuc}$ to the objective function makes the objective "more convex", but not necessarily convex. The use of the nuclear norm has particular relevance as it is closely associated with minimizing the rank of Z [18], which corresponds to the number of clusters. As $\alpha \geq 0$ increases from zero, the number of clusters represented by Z should decrease. If $\alpha \to \infty$, then eventually there is only one cluster.

3. IMPLEMENTATION AND COMPUTATIONAL RESULTS

3.1 Implementation

The optimization task was implemented by a gradient-projection method. For a step length s, steps are taken in the directions of the negative gradients of $Z \mapsto D \bullet \left(ZZ^T\right)$ and $Z \mapsto \alpha\|Z\|_{nuc}$, and then the result is projected onto the nearest point in the feasible set: $\left\{ Z \mid Z \geq 0 \text{ and } Ze^{(K)} = e^{(N)} \right\}$. The distance measure used was the Frobenius norm $\|Z\|_F = \sqrt{Z \bullet Z} = \left[\sum_{i,j} z_{ij}^2\right]^{1/2}$. The gradient direction of the nuclear norm $Z \mapsto \|Z\|_{nuc}$ can be computed by means of a reduced singular value decomposition (SVD). If $Z = U\Sigma V^T$ is the reduced SVD (U is $N \times K$ and V is $K \times K$, both with orthonormal colums, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_K)$), then the gradient step is $\delta Z = -s\,\alpha\,UV^T$, which is equivalent to setting $\sigma_i \leftarrow \sigma_i - \alpha\,s$. In order to avoid negative singular values, negative singular values are replaced by zero: $\sigma_i \leftarrow \max(\sigma_i - \alpha\,s, 0)$, for $i = 1, 2, \ldots, K$.

The projection onto the feasible set amounts to projecting each row of Z onto the unit simplex $\left\{ \mathbf{x} \mid \mathbf{x} \geq 0 \text{ and } \mathbf{e}^T\mathbf{x} = 1 \right\}$. This can be done using the algorithm in [11].

These gradient and projection steps are repeated for a user-specified number of steps.

To compare two clusterings with the same number of clusters, we need to identify an optimal matching between the clusters of each clustering. So for clusterings $C = \{C_1, C_2, \ldots, C_K\}$ and $D = \{D_1, D_2, \ldots, D_K\}$ where the clusters C_i and D_j are subsets of the data items $\{1, 2, \ldots, N\}$, we need to find a permuation π so that cluster C_i is matched to cluster D_j with $j = \pi(i)$. To do this in an optimal way, we first create a matrix $B = [b_{ij}]$ where b_{ij} is the number of data items in $C_i \cap D_j$. We then want to find a permutation matrix P that maximizes trace(BP), which is the sum of the diagonal entries of BP. This task is equivalent to a linear programming problem [3] because the objective function is linear and the convex hull of $N \times N$ permutation matrices is the set of $N \times N$ doubly stochastic matrices $\left\{ P \in \mathbb{R}^{N \times N} \mid P \geq 0,\ Pe^{(N)} = e^{(N)},\ P^T e^{(N)} = e^{(N)} \right\}$. Thus this task can be accomplished by a simplex-type algorithm which swaps pairs of columns.

The method was implemented using MATLAB® 7.6.1.

3.2 Computational results

Computational results were obtained for a rat central nervous system study (CNS-Rat) [20].

The algorithm was run with on the CNS-Rat data set with $d_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\|_2^2$ using the normalized values (maximum levels of gene expression set to one for each gene), and $\alpha = 10$, and a step size $s = 5 \times 10^{-3}$. Significantly larger values of s, such as $s = 10^{-2}$, resulted in the objective function increasing rather than decreasing. Several initial values of Z were generated, and 1000 gradient projection steps were carried out. The computed Z were then "quantized" by setting $\widehat{z}_{ij} = 1$ if $z_{ij} = \max_k z_{ik}$, and $\widehat{z}_{ik} = 0$ if $k \neq j$, for each i. The clusters so computed were then compared with the published clusters given in [20]. These published clusters were obtained using the Fitch–Margoliash method [5] for creating phylogenetic trees using a least-squares technique. Twenty runs were carried out for this data set with different randomized starting values for Z with the entries being pseudorandom numbers in the range $[0, 1]$. Each row of these matrices was projected onto the unit simplex to ensure feasibility. Objective function values were recorded for the computed Z matrices, and for the "quantized" Z matrices.

Each run took about 30 to 40 seconds.

3.2.1 *Multiple local minima*

From the diagnostic output it appeared that the gradient projection method had approached a local minimum or had stagnated in each run. The computed matrices Z typically had all but about six to twelve entries that were not zero or one; some of these were clearly close to zero or one. A serious concern about this algorithm is that the computed matrices Z converge to points that are local minima but far from a global minimum. The runs show evidence that there are significant local minima that are not global minima, as can be seen in Table 1.

In spite of the fact of multiple local minima, only a few runs are required to obtain consistent values. The computed Z giving the best computed objective value is denoted Z^*. The quantized version of Z^* is $\widehat{Z}^*$. As we might hope, the objective function values for the quantized matrices $\widehat{Z}^*$ is only slightly above the computed Z^*.

Run	$\phi(Z^*)$	$\phi(\widehat{Z}^*)$	Run	$\phi(Z^*)$	$\phi(\widehat{Z}^*)$
1	**1531.84**	**1532.57**	11	1540.73	1542.23
2	1540.65	1542.17	12	**1531.84**	**1532.57**
3	1534.43	1535.63	13	1534.49	1536.11
4	1566.70	1567.32	14	1534.64	1534.81
5	*1532.14*	*1532.86*	15	1569.72	1572.10
6	**1531.84**	**1532.57**	16	**1531.84**	**1532.57**
7	*1532.14*	*1532.86*	17	**1531.84**	**1532.57**
8	1533.89	1535.92	18	*1532.14*	*1532.86*
9	1533.26	1533.52	19	*1532.14*	*1532.86*
10	*1532.14*	*1532.86*	20	**1531.84**	**1532.57**

Table 1: Objective function values $\phi(Z) :=$ trace$(Z^T W Z) + \alpha \|Z\|_{nuc}$ for runs with different starting points for computed matrices Z^* and the quantized matrices $\widehat{Z}^*$. Minimum values computed are in bold; 2nd smallest are in italics.

Run	best computed	published	Run	best computed	published
1	0	27	11	6	29
2	7	28	12	0	27
3	4	29	13	3	27
4	13	28	14	8	32
5	2	28	15	19	25
6	0	27	16	0	27
7	2	28	17	0	27
8	1	27	18	2	28
9	3	29	19	2	28
10	2	28	20	0	27

Table 2: Mismatches between computed clusterings and the best computed clustering and the published clustering.

3.2.2 *Comparison with the clusterings obtained and the published clustering*

The objective function values given in Table 1 are well below the objective function value for the published clustering: $\phi(Z_{pub}) \approx 2130.82$. Even if we ignore the nuclear norm part of the objective function we see that trace$(Z_{pub}^T W Z_{pub}) \approx 1876.76$, which is still much higher than the values computed for $\phi(\widehat{Z}^*)$ which includes this term.

The computed quantized matrices tend to be very consistent in terms of the clusterings they produce, as can be seen in the mismatches between the different computed clusterings (being the number of incorrectly clustered data items) as shown in Table 2.

The disagreements between the computed and "exact" clusterings appear to be fairly consistent, at least in terms of numbers of mismatches. Another claimed clustering provided with the data had 62 mismatches with the "exact" clustering; this clustering had 60 mismatches with the best computed clustering.

4. CONCLUSIONS

This paper presents a matrix-based optimization method for clustering data that is suitable for data sets of biological interest. These methods are practical and appears capable of accurately clustering data sets. A number of refinements are possible to improve computation times. The step length can be adjusted during the computation, which should aid convergence, and the code can be modified

to improve the execution time. Implementation in compiled rather than interpreted languages can also dramatically reduce the execution time.

5. REFERENCES

[1] Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–120, 2002.

[2] Steven J. Benson, Yinyu Ye, and Xiong Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461, 2000.

[3] G. B. Dantzig. *Linear Programming and Extensions*. Princetn Uni. Press, Princeton, NJ, 1963.

[4] V. Faber. Clustering and the continuous k-means algorithm. Technical Report LA_Science_22n15, Los Alamos National Laboratory, November 1994.

[5] Walter M. Fitch and Emanuel Margoliash. Construction of phylogenetic trees. *Science*, 155(3760):279–284, 1967.

[6] Anil K. Ghosh. On optimum choice of k in nearest neighbor classification. *Comput. Statist. Data Anal.*, 50(11):3113–3123, 2006.

[7] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Math. Programming*, 79(1-3, Ser. B):191–215, 1997. Lectures on mathematical programming (ISMP97) (Lausanne, 1997).

[8] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.

[9] Masakazu Kojima, Susumu Shindoh, and Shinji Hara. Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM J. Optim.*, 7(1):86–125, 1997.

[10] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28(2):129–137, 1982.

[11] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbf{R}^n$. *J. Optim. Theory Appl.*, 50(1):195–200, 1986.

[12] S. T. Milagre, C. D. Maciel, A. A. Shinoda, M. Hungria, and J. R. B. Almeida. Multidimensional cluster stability analysis from a Brazilian *bradyrhizobium sp.* RFLP/PCR data set. *J. Comput. Appl. Math.*, 227(2):308–319, 2009.

[13] S. Oliveira, T. Soma, and D. Stewart. Semidefinite programming for graph partitioning with preference in data distribution. In *5th International Conference High Performance Computing for Computational Science - VECPAR 2002*, volume 2565 of *Lecture Notes in Computer Science*, pages 307–321, Heidelberg, 2003. Springer.

[14] James Joseph Palmersheim. *Nearest Neighbor Classification Rules: Small Sample Performance and Comparison with the Linear Discriminant Function and the Optimum Rule*. ProQuest LLC, Ann Arbor, MI, 1970. Thesis (Ph.D.)–University of California, Los Angeles.

[15] Javier Peña, Juan Vera, and Luis F. Zuluaga. Computing the stability number of a graph via linear and semidefinite programming. *SIAM J. Optim.*, 18(1):87–105 (electronic), 2007.

[16] Jiming Peng and Yu Wei. Approximating K-means-type clustering via semidefinite programming. *SIAM J. Optim.*, 18(1):186–205 (electronic), 2007.

[17] M. R. Rao. Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66(335):622–626, 1971.

[18] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010.

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

[20] XL Wen, S Fuhrman, GS Michaels, DB Carr, S Smith, JL Barker, and R Somogyi. Large-scale temporal gene expression mapping of central nervous system development. *Proceedings of the National Academy of Sciences of the United States of America*, 95(1):334–339, JAN 6 1998.

[21] Gordon Wilfong. Nearest neighbor problems. *Internat. J. Comput. Geom. Appl.*, 2(4):383–416, 1992.

[22] Henry Wolkowicz. Semidefinite and Lagrangian relaxations for hard combinatorial problems. In *System modelling and optimization (Cambridge, 1999)*, pages 269–309. Kluwer Acad. Publ., Boston, MA, 2000.

[23] Henry Wolkowicz and Miguel F. Anjos. Semidefinite programming for discrete optimization and matrix completion problems. *Discrete Appl. Math.*, 123(1-3):513–577, 2002. Workshop on Discrete Optimization, DO'99 (Piscataway, NJ).

[24] Henry Wolkowicz and Qing Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Appl. Math.*, 96/97:461–479, 1999. The satisfiability problem (Certosa di Pontignano, 1996); Boolean functions.

[25] Xiang Yin and Alex Tay Leng Phuan. Genetic algorithm based K-means fast learning artificial neural network. In *AI 2004: Advances in artificial intelligence*, volume 3339 of *Lecture Notes in Comput. Sci.*, pages 828–839. Springer, Berlin, 2004.