Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2015, Article ID 196098, 10 pages
http://dx.doi.org/10.1155/2015/196098

Research Article

A Fast Clustering Algorithm for Data with a Few Labeled Instances

Jinfeng Yang,[1] Yong Xiao,[1] Jiabing Wang,[2] Qianli Ma,[2] and Yanhua Shen[3]

[1]*Electric Power Research Institute of Guangdong Power Grid Corporation, Guangzhou 510080, China*
[2]*School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China*
[3]*School of Materials Science and Engineering, South China University of Technology, Guangzhou 510006, China*

Correspondence should be addressed to Jiabing Wang; jbwang@scut.edu.cn

Received 1 November 2014; Revised 27 February 2015; Accepted 1 March 2015

Academic Editor: Pietro Aricò

The diameter of a cluster is the maximum intracluster distance between pairs of instances within the same cluster, and the split of a cluster is the minimum distance between instances within the cluster and instances outside the cluster. Given a few labeled instances, this paper includes two aspects. First, we present a simple and fast clustering algorithm with the following property: if the ratio of the minimum split to the maximum diameter (RSD) of the optimal solution is greater than one, the algorithm returns optimal solutions for three clustering criteria. Second, we study the metric learning problem: learn a distance metric to make the RSD as large as possible. Compared with existing metric learning algorithms, one of our metric learning algorithms is computationally efficient: it is a linear programming model rather than a semidefinite programming model used by most of existing algorithms. We demonstrate empirically that the supervision and the learned metric can improve the clustering quality.

1. Introduction

Clustering is the unsupervised classification of instances into clusters in a way that attempts to minimize the intracluster distance and to maximize the intercluster distance. Two criteria commonly used to measure the quality of a clustering are diameter and split. The diameter of a cluster is the maximum distance between pairs of instances within the same cluster, and the split of a cluster is the minimum distance between instances within the cluster and instances outside the cluster. Clearly, the diameter of a cluster is a natural indication of homogeneity of the cluster and the split of a cluster is a natural indication of separation between the cluster and other clusters.

Many authors studied optimization problems related to the diameter or the split of cluster, for example, to minimize the maximum cluster diameter [1–4]; minimize the sum of cluster diameters or radii [5–8]; or maximize the ratio of the minimum split to the maximum diameter [9]. The well-known single-linkage clustering and the complete-linkage clustering also optimize the two criteria, respectively: the former maximizes the minimum cluster split, and the later attempts to minimize the maximum cluster diameter.

Ackerman and Ben-David [10] defined a set of axioms that a measure of cluster-quality should satisfy *scale invariance*, *isomorphism invariance*, *weak local consistency*, and *cofinal richness*, and they showed that the RSD clustering criterion, that is, maximizing of the ratio of the minimum split to the maximum diameter, satisfies those axioms. Given data X, let $\mathrm{RSD}_{\mathrm{opt}}(X)$ be the maximum RSD of X among all possible partitions of X into k clusters. If $\mathrm{RSD}_{\mathrm{opt}}(X) > 1$, the optimal solution with respect to the RSD criterion has the following property: the distance between each pair of instances in different clusters is larger than that of each pair of instances within the same cluster. Hence, we say that data X is *well-clusterable* if $\mathrm{RSD}_{\mathrm{opt}}(X) > 1$, and X' are *more clusterable* than X if $\mathrm{RSD}_{\mathrm{opt}}(X') > \mathrm{RSD}_{\mathrm{opt}}(X)$.

Ackerman and Ben-David [11] showed that if $\mathrm{RSD}_{\mathrm{opt}}(X) > 1$, then the optimal solution with respect to the RSD criterion can be found in time $O(n^2 \log n)$, where n is the number of instances in X. In this paper, we further show that if $\mathrm{RSD}_{\mathrm{opt}}(X) > 1$, then the optimal solution with the following

criteria can be found using Gonzalez's algorithm [1] in linear time: maximizing RSD, maximizing the minimum split, and minimizing the maximum diameter.

However, the condition of $\mathrm{RSD}_{\mathrm{opt}}(X) > 1$ is too strong and unrealistic for real world data. So, a natural problem arises if X is poorly clusterable ($\mathrm{RSD}_{\mathrm{opt}}(X) \ll 1$), whether X can be made more clusterable by a metric learning approach and thus Gonzalez's algorithm together with the learned metric can perform better than together with the original metric.

In the clustering literature, there are commonly two methods to add supervision information into clustering. First, adding a small portion of the training data into unlabeled data, this method is also called semisupervised learning [12, 13]. Second, instead of specifying the class labels, pairwise constraints are specified [14, 15]: a pairwise must-link constraint corresponds to the requirement that the involved two instances must be within the same cluster, whereas the two instances involved in a cannot-link constraint must be in different clusters.

Metric learning can be grouped into two categories, that is, unsupervised and supervised metric learning. In this paper, we focus on supervised metric learning. Supervised metric learning attempts to learn distance metrics that keep instances with the same class label (or with a must-link constraint) close and separate instances with different class labels (or with a cannot-link constraint) far away. Since there are many possible ways to realize this intuition, a great number of algorithms have been developed for supervised metric learning, for example, Local Linear Discriminative Analysis (LLDA) [16], Relevant Components Analysis (RCA) [17], Xing et al.'s algorithm [18], Locally Linear Metric Adaptation (LLMA) [19], Neighborhood Component Analysis (NCA) [20], Discriminative Component Analysis (DCA) [21], Local Fisher Discriminant Analysis (LFDA) [22], Large Margin Nearest Neighbor (LMNN) [23], Local Distance Metric (LDM) [24], Information-Theoretic Metric Learning (ITML) [25], Laplacian Regularized Metric Learning (LRML) [26], Generalized Sparse Metric Learning (GSML) [27], Sparse Distance Metric Learning (SDML) [28], Multi-Instance MEtric Learning (MIMEL) [29], online-reg [30], Constrained Metric Learning (CML) [31], mixture of sparse Neighborhood Components Analysis (msNCA) [32], Metric Learning with Multiple Kernel Learning (ML-MKL) [33], Least Squared residual Metric Learning (LSML) [34], and Distance Metric Learning with eigenvalue (DML-eig) [35].

Overall, empirical studies showed that supervised metric learning algorithms can usually outperform unsupervised ones by exploiting either the label information or the side information presented in pairwise constraints. However, despite extensive studies, most of the existing algorithms for metric learning have one of the following drawbacks: it needs to solve a nontrivial optimization problem, for example, a semidefinite programming problem, there are parameters to tune, and the solution is local optimal.

In this paper, we present two simple metric learning models to make data more clusterable. The two models are computationally efficient, parameter-free, and local-optimality-free. The rest of this paper is organized as follows. Section 2 gives some notations and the definitions of clustering criteria used in the paper. Section 3 gives Gonzalez's farthest-point clustering algorithm for unsupervised learning, presents a nearest neighbor-based clustering algorithm for the semi-supervised learning, and discusses the properties of the two algorithms. In Section 4, we formularize the problem of making data more clusterable as a convex optimization problem. Section 5 presents the experimental results. We conclude the paper in Section 6.

2. Notations and Preliminary

We use the following notations in the rest of the paper.

$|\cdot|$: the cardinality of a set.

$X \subset \mathfrak{R}^d$: the set of instances (in d-dimension space) to be clustered.

$d(x, y)$: the Euclidian distance between $x \in X$ and $y \in X$.

$S_1, S_2, \ldots, S_k$: the k small subsets of X with given labels, that is, the supervision. In this paper, we assume that either $S_i \neq \Phi$ for $i = 1, 2, \ldots, k$ (the case of semisupervised learning) or $S_i = \Phi$ for $i = 1, 2, \ldots, k$ (the case of unsupervised learning).

$\wp$: the set of all partitions of n objects into k nonempty and disjoint clusters $\{C_1, C_2, \ldots, C_k\}$.

Definition 1. Given $S_1, S_2, \ldots, S_k$, we say that a partition $P \in \wp$ respects the semi-supervised constraints if P satisfies the following conditions.

(1) All instances in S_i must be within the same cluster of P for $i = 1, 2, \ldots, k$, and

(2) Any pair of instances $x \in S_i$ and $y \in S_j$, x and y must be in different clusters of P for $i, j = 1, 2, \ldots, k$, and $i \neq j$.

In the rest of the paper, we use $\wp_{\mathrm{ssc}}$ to denote the subset of $\wp$ that respects the semisupervised constraints, and we require that any partition in the context of semisupervised learning should respect the semisupervised constraints.

Definition 2. For a set C of objects, the split $s(C)$ of C is defined as

$$s(C) = \min_{x \in C, y \notin C} d(x, y). \tag{1}$$

For a partition $P = \{C_1, C_2, \ldots, C_k\} \in \wp$, the split $s(P)$ of P is the minimum $s(C_i)$ among $i = 1, 2, \ldots, k$.

Definition 3. For a set C of objects, the diameter $d(C)$ of C is defined as

$$d(C) = \max_{x, y \in C} d(x, y). \tag{2}$$

For a partition $P = \{C_1, C_2, \ldots, C_k\} \in \wp$, the diameter $d(P)$ of P is the maximum diameter $d(C_i)$ of C_i among $i = 1, 2, \ldots, k$.

Definition 4. The unsupervised and semisupervised max-min split problems are defined as, respectively,

$$\max_{P\in\wp} s(P), \tag{3}$$

$$\max_{P\in\wp_{\text{scc}}} s(P). \tag{4}$$

Definition 5. The unsupervised and semisupervised min-max diameter problems are defined as, respectively,

$$\min_{P\in\wp} d(P), \tag{5}$$

$$\min_{P\in\wp_{\text{scc}}} d(P). \tag{6}$$

Definition 6. The unsupervised and semisupervised max-RSD problems are defined as, respectively,

$$\max_{P\in\wp} \frac{s(P)}{d(P)}, \tag{7}$$

$$\max_{P\in\wp_{\text{scc}}} \frac{s(P)}{d(P)}. \tag{8}$$

For the unsupervised max-RSD problem, Wang and Chen [9] presented an exact algorithm for $k = 2$ and a 2-approximation algorithm for $k \geq 3$; however the worst-case time complexity of both algorithms is $O(n^3)$ and thus impractical for large-scale data.

Let $S \subseteq X$; we use $d\,\text{Max}(x, S)$ to denote the maximum distance between the instance x and instances in S; that is, $d\,\text{Max}(x, S) = \max\{d(x, y) \mid y \in S\}$; similarly, $d\,\text{Min}(x, S) = \min\{d(x, y) \mid y \in S\}$.

3. Well-Clusterable Data: Find the Optimal Solution Efficiently

In this section, we show that if $\text{RSD}_{\text{opt}}(X) > 1$, the max-RSD problem, the max-min split problem, and the min-max diameter problem can be simultaneously solved by Gonzalez's algorithm for unsupervised learning in Section 3.1 and by a nearest neighbor-based algorithm for semisupervised learning in Section 3.2, respectively. At the same time, we also discuss the properties of the two algorithms for the case of $\text{RSD}_{\text{opt}}(X) \leq 1$.

3.1. Unsupervised Learning. The farthest-point clustering (*FPC*) algorithm proposed by Gonzalez [1] is shown in Algorithm 1, where the meaning of nearest neighbor is its literal one as (9); that is, p's nearest neighbor in R is q,

$$q = \arg\min_{u\in R} d(p, u). \tag{9}$$

Theorem 7. *For unsupervised learning, if $RSD_{opt}(X) > 1$, then the partition P returned by FPC is simultaneously the optimal solution of the max-RSD problem, the max-min split problem, and the min-max diameter problem.*

Algorithm: *FPC*
Input: The input data X, and the number k of clusters.
Output: The partition P of X.
$R \leftarrow \Phi$;
Randomly select an instance p from X;
$R \leftarrow R \cup \{p\}$;
while ($|R| < k$)
  $p \leftarrow \arg\max_{q\in X-R} d\,\text{Min}(q, R)$;
  $R \leftarrow R \cup \{p\}$;
end while
Let P be the partition by assigning each instance p of X to its nearest neighbor in R (if $p \in R$, the nearest neighbor of p in R is itself);
return P;

Algorithm 1: The *FPC* algorithm for unsupervised learning [1].

Proof. (a) The proof of the max-RSD problem: let $P' = \{C_1, C_2, \ldots, C_k\}$ be the optimal partition of the max-RSD problem; then $\text{RSD}(P') > 1$, and we have

$$\forall p, q \in C_i,\ \forall u \notin C_i:\quad d(p, q) < d(p, u) \quad \forall i. \tag{10}$$

We prove the following proposition: any pair of instances in R (see Algorithm 1) must be in different clusters of P'; that is, R contains exactly one instance of each cluster C_i, $i = 1, 2, \ldots, k$. If this holds, then by (10), for any instance $q \in C_i$, $i = 1, 2, \ldots, k$, its nearest neighbor in R must be the instance p such that p also belongs to C_i, and hence $P = P'$.

We prove the proposition by contradiction. Assume that there exists a pair of instances p and q in R so that they belong to the same cluster C_r for some r. Without loss of generality, let p be selected into R before q. Then $d\,\text{Min}(q, R) \leq d(q, p)$ when selecting q into R. Note that $|R| < k$ before selecting q; there exists at least one cluster C_t ($t \neq r$) such that no instance in C_t belongs to R. By (10), for any $q' \in C_t$, we have $d\,\text{Min}(q', R) > d(q, p) \geq d\,\text{Min}(q, R)$; q has no chance to be selected into R since we should select the instance q' with the maximum $d\,\text{Min}(q', R)$, and thus the proposition holds.

(b) Since separating any pair p, q of instances within the same cluster of P into different clusters will strictly decrease the split of the resulted partition, the conclusion for the max-min split problem holds.

(c) Since grouping any pair p, q of instances in different clusters of P into the same cluster will strictly increase the diameter of the resulting partition, the conclusion for the min-max diameter problem holds. □

Clearly, the time complexity of *FPC* is $O(nk)$ by maintaining a nearest neighbor table that records the nearest neighbor in R of each instance $p \in X - R$ and the corresponding distance between p and its nearest neighbor in R. The space complexity is $O(n)$. So, the time complexity and the space complexity are both linear with n for a fixed k. Using a more complicated approach, the *FPC* algorithm can be implemented in $O(n \log k)$, but the implementation was exponentially dependent on the dimension d [3].

Algorithm: *NNC*
Input: The input data X, the number k of clusters, and the k labeled subsets $S_1, S_2, \ldots, S_k$ of X.
Output: The partition P of X.
for each unlabelled instance $p \in X$, compute $d\,\mathrm{Max}(p, S_i)$ for $i = 1, 2, \ldots, k$;
Let $C_i = S_i$ for $i = 1, 2, \ldots, k$;
for each unlabelled instance $p \in X$
  $r \leftarrow \arg\min_{i \in \{1,2,\ldots,k\}} d\,\mathrm{Max}(p, S_i)$;
  $C_r \leftarrow C_r \cup \{p\}$;
end for
return $P = \{C_1, C_2, \ldots, C_k\}$;

Algorithm 2: The *NNC* clustering algorithm for semisupervised learning.

Now, a natural problem arises: if $\mathrm{RSD}_{\mathrm{opt}}(X) \leq 1$, how does the *FPC* algorithm perform? Although, in this paper, we cannot give performance guarantee of the *FPC* algorithm for the max-RSD problem and the max-min split problem if $\mathrm{RSD}_{\mathrm{opt}}(X) \leq 1$, Gonzalez [1] proved the following theorem (see also [2, 3]).

Theorem 8 (see [1]). *The FPC is a 2-approximation algorithm for the unsupervised min-max diameter problem with the triangle inequality satisfied for any k. Furthermore, for $k \geq 3$, the $(2 - \varepsilon)$-approximation of the unsupervised min-max diameter problem with the triangle inequality satisfied is NP-complete for any $\varepsilon > 0$.*

So as far as the approximation ratio is concerned, the FPC algorithm is the best for the unsupervised min-max diameter problem unless P = NP.

3.2. Semi-Supervised Learning. For semisupervised learning, we present a nearest neighbor-based clustering (*NNC*) algorithm as shown in Algorithm 2. The algorithm is self-explanatory, and we do not give a further explanation.

Theorem 9. *For semiunsupervised learning, if $RSD_{opt}(X) > 1$, then the partition P returned by NNC is simultaneously the optimal solution of the semisupervised max-RSD problem, the semisupervised max-min split problem, and the semisupervised min-max diameter problem.*

Proof. The proof of max-RSD(P) problem: let $P' = \{C_1', C_2', \ldots, C_k'\}$ be the optimal partition of the semisupervised max-RSD problem. Since P' respects the supervision, we can replace S_i by a super-instance α_i for $i = 1, 2, \ldots, k$; then each cluster C_i' contains exactly one super-instance α_i for $i = 1, 2, \ldots, k$ (without loss of generality, here we assume that α_i is in the cluster C_i' for $i = 1, 2, \ldots, k$). Let $P = \{C_1, C_2, \ldots, C_k\}$; then according to the algorithm *NNC*, each cluster also contains exactly one super-instance, and without loss of generality, we also assume that α_i is in the cluster C_i for $i = 1, 2, \ldots, k$. For each unlabeled instance $p \in C_r'$ for $r = 1, 2, \ldots, k$, since $\mathrm{RSD}_{\mathrm{opt}}(X) > 1$, we have $d(p, \alpha_r) = d\,\mathrm{Max}(p, S_r) < d(p, \alpha_i) = d\,\mathrm{Max}(p, S_i)$ for any $i \neq r$, and the nearest neighbor of p in $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ is α_r, so $C_r' = C_r$ for $r = 1, 2, \ldots, k$, and thus $P' = P$.

The proofs for the semisupervised max-min split problem and the semisupervised min-max problem are similar to (b) and (c) in the proof of Theorem 7 respectively, and here we omit it. □

The time complexity of *NNC* using a simple implementation is

$$\sum_{i=1}^{k} O\left(n\left|S_i\right|\right) + O(nk). \tag{11}$$

The space complexity of *NNC* is $O(n)$. Since we assume that S_i are small sets for $i = 1, 2, \ldots, k$, the time and space complexities are also linear with n when $|S_i|$ are regarded as constants for $i = 1, 2, \ldots, k$.

Similar to Theorem 8, we have the following theorem for the semisupervised min-max diameter problem.

Theorem 10. *NNC is a 2-approximation algorithm for the semisupervised min-max diameter problem with the triangle inequality satisfied.*

Proof. Let $S = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ (see the proof of Theorem 9), $\delta = \max\{d(S_1), d(S_2), \ldots, d(S_k)\}$, and $\sigma = \max\{d\,\mathrm{Min}(q, S) \mid q$ is an unlabelled instance$\}$ and let p be any unlabelled instance such that $d\,\mathrm{Min}(p, S) = \sigma$. Since the optimal partition of a semisupervised min-max diameter problem must respect the supervision, we have $d_{\mathrm{opt}}(X) \geq \delta$, where $d_{\mathrm{opt}}(X)$ denotes the diameter of the optimal solution of the semisupervised min-max diameter problem; at the same time, p and α_i for some $i \in \{1, 2, \ldots, k\}$ must be within the same cluster of the optimal solution, so $d_{\mathrm{opt}}(X) \geq \sigma$; therefore $d_{\mathrm{opt}}(X) \geq \max\{\delta, \sigma\}$. Now consider the partition $P = \{C_1, C_2, \ldots, C_k\}$ returned by *NNC*. Since each unlabeled instance q is assigned into its nearest neighbor in S, so, for any cluster C_i of P for $i = 1, 2, \ldots, k$ (assume that the super-instance in C_i is α_i), we have $d(q, \alpha_i) \leq \sigma$, and $d(C_i) \leq 2\sigma$ by the triangle equality. So, $d(P) \leq \max\{2\sigma, \delta\} \leq 2d_{\mathrm{opt}}(X)$, and the theorem holds. □

4. The Metric Learning Models

If the given data are poorly clusterable, that is, the $\mathrm{RSD}_{\mathrm{opt}}(X)$ is far less than one, the algorithms *FPC* and *NNC* may perform poorly. Given the supervision, we use metric learning to make the supervised data more clusterable, and then the two algorithms can be used with the new metric.

Supervised metric learning attempts to learn distance metrics that keep instances with the same class labels (or with a must-link constraint) close and separate instances with different class labels (or with a cannot-link constraint) far away. As discussed in the first section, there are many possible

ways to realize this intuition; for example, Xing et al. [18] presented the following model:

$$\min_{M} \quad \sum_{(x,y)\in S} \|x-y\|_M^2 \tag{12}$$

$$\text{s.t.} \quad \sum_{(x,y)\in D} \|x-y\|_M \geq 1 \tag{13}$$

$$M \succeq 0. \tag{14}$$

In the above model, S denotes the set of must-link constraints, D denotes the set of cannot-link constraints, M is a $d\times d$ Mahalanobis distances matrix, and $\|x-y\|_M$ denotes the distance $d(x, y)$ between two instances x and $y \in X \subseteq \Re^d$ with respect to M; that is,

$$\|x-y\|_M = \sqrt{(x-y)^T M (x-y)}, \tag{15}$$

where T denotes the transpose of a matrix or a vector. The constraint (14) requires that M should be a positive semidefinite matrix; that is, $\forall x \in \Re^d$, $x^T M x \geq 0$. The choice of the constant 1 on the right hand side of (13) is arbitrary but not important, and changing it to any other positive constant c results only in M being replaced by $c^2 M$.

Note that the matrix M can be either a full matrix or a diagonal matrix. In natural language, Xing et al.'s model minimizes the sum of the square of distance with respect to M between pairs of instances with must-link constraints subject to the following constraints: (a) the sum of distances with respect to M between pairs of instances with cannot-link constraints is greater than or equal to one, and (b) M is a positive semidefinite matrix.

Xing et al.'s model, as well as most of the existing metric learning, is a semidefinite programming problem and thus computationally expensive and even intractable in high dimensional space for the case of full matrix.

Inspired by the RSD clustering criterion, we propose two metric learning models: one learns a full matrix and the other learns a diagonal matrix. In this section, the supervision can be given either in the form of labeled sets $S_1, S_2, \ldots, S_k$ or in the form of pairwise constraints.

4.1. The Labeled Sets. Given the supervision $S_1, S_2, \ldots, S_k$, we want to learn a Mahalanobis distances matrix M such that the minimum split with respect to M among S_i, $i = 1, 2, \ldots, k$, is maximized subject to the following constraints: (a) the distance between each pair of instances with the same class label is less than or equal to one and (b) M is a positive semidefinite matrix. Formally, we have the following optimization problem (the case of full matrix).

The Case of Full Matrix. Consider

$$\max_{M} \quad s \tag{16}$$

$$\text{s.t.} \quad \forall x \in S_i,\ y \in S_j: \quad \|x-y\|_M \geq s, \quad i, j = 1, 2, \ldots, k, \quad i \neq j \tag{17}$$

$$\forall x, y \in S_i: \quad \|x-y\|_M \leq 1, \quad i = 1, 2, \ldots, k \tag{18}$$

$$M \succeq 0 \tag{19}$$

$$s \geq 0. \tag{20}$$

The constraint (17) requires that the scalar variable s (the minimum split) is the minimum among distances between pairs of instances with different class labels. The constraint (18) requires that the distance between each pair of instances with the same class label is less than or equal to one. The optimization objective is to maximize s. Similar to (13), the choice of the constant 1 on the right hand side of (18) is arbitrary but not important and can be set to any positive constant.

The full matrix model is a SDP optimization problem, and, theoretically, the global optimal solution can be solved efficiently [36]. However, when M is a full matrix, the number of variables ($|M|$) is quadratic in d, and thus it is prohibitive for problems with a large number of dimensions. To avoid this problem, we can require that M is a diagonal matrix. Since M is a diagonal matrix, M is a positive semidefinite matrix if and only if $M_{ii} \geq 0$ for $i = 1, 2, \ldots, d$, where M_{ii} is the ith diagonal entry. So, learning a diagonal matrix M is equivalent to learning a vector $z \in \Re^d$ using the following model (the case of diagonal matrix).

The Case of Diagonal Matrix. Consider

$$\max_{z} \quad s \tag{21}$$

$$\text{s.t.} \quad \forall x \in S_i,\ y \in S_j: \quad \|x-y\|_z \geq s, \quad i, j = 1, 2, \ldots, k, \quad i \neq j \tag{22}$$

$$\forall x, y \in S_i: \quad \|x-y\|_z \leq 1, \quad i = 1, 2, \ldots, k \tag{23}$$

$$z \geq 0 \tag{24}$$

$$s \geq 0, \tag{25}$$

where

$$\|x-y\|_z = \sqrt{\sum_{i=1}^{d} z_i (x_i - y_i)^2}. \tag{26}$$

The constraint (24) requires that each component of z should be greater than or equal to zero.

Now since the optimization objective and all constraints are linear, the above optimization problem is a linear programming problem with $d + 1$ variables, and $k \times |S_i| \times (|S_i| - 1)/2 + k\times(k-1)\times|S_i|^2/2 + (d+1)$ inequality constraints (assume

that S_i has equal size). When $|S_i|$ is small for $i = 1, 2, \ldots, k$, the global optimal solution can be efficiently found using some optimization tool package, for example, the MATLAB *linprog* function, or the CVX—MATLAB software for disciplined convex programming (http://cvxr.com/cvx/download/).

4.2. Pairwise Constraints. If the supervision is given in the form of pairwise constraints, that is, the *must-link* and *cannot-link* constraints, the models also work after a minor modification. Let ML be the set of must-link constraints, and let CL be the set of cannot-link constraints; then the full matrix model and the diagonal matrix model should be modified as follows: substituting (17′) for (17), (18′) for (18), (22′) for (22), and (23′) for (23), respectively,

$$\forall (x, y) \in \text{ML}: \quad \|x - y\|_M \leq 1, \tag{17'}$$

$$\forall (x, y) \in \text{CL}: \quad \|x - y\|_M \geq s, \tag{18'}$$

$$\forall (x, y) \in \text{ML}: \quad \|x - y\|_z \leq 1, \tag{22'}$$

$$\forall (x, y) \in \text{CL}: \quad \|x - y\|_z \geq s. \tag{23'}$$

However, if the supervision is given in the form of pairwise constraints, it is nontrivial to decide whether there is a partition P of X such that P satisfies all of those pairwise constraints (and we call it the feasibility problem). For CL constraints, Davidson and Ravi showed that the feasibility problem is equivalent to the k-colorability problem [37] and thus NP-complete [38], whereas the feasibility problem is trivial if the supervision is given in the form of labeled sets. Of course, if we do not require that all of those pairwise constraints should be satisfied, the *FPC* algorithm can be naturally used together with the metric learned from the pairwise constraints.

Clearly, the metric learning models proposed in this paper are practicable only when the cardinality of sets of labeled instances or the number of pairwise constraints is small. Otherwise, the problem is usually overconstrained and there is no feasible solution.

5. The Experimental Results

5.1. The Compared Algorithms and Benchmark Datasets. To validate whether semisupervised learning performs better than unsupervised one, whether metric learning can improve clustering quality, and whether our metric learning model performs better than Xing et al.'s one for the *FPC* and *NNC* algorithms, we implemented the following algorithms:

(i) the *FPC* algorithm as shown in Algorithm 1;

(ii) the *NNC* algorithm as shown in Algorithm 2;

(iii) the *FPC* with our metric learning model (the case of diagonal matrix) (*FPC_Diag*); that is, we first use our metric learning model to learn a vector z and then use the *FPC* clustering algorithm with the learned vector; that is, the distance is computed using (26);

(iv) the *NNC* with our metric learning model (the case of diagonal matrix) (*NNC_Diag*);

(v) the *FPC* with Xing et al.'s metric learning algorithm (also using the diagonal matrix) (*FPC_Xing*); that is, we first use Xing et al.'s metric learning algorithm to learn a vector z and then use the *FPC* clustering algorithm with the learned vector;

(vi) the *NNC* with Xing et al.'s metric learning algorithm (also using the diagonal matrix) (*NNC_Xing*).

We also implemented the following algorithms as baseline approaches. The reason that we select *k-means* to compare is that *k-means* is very simple and also a linear time algorithm when regarding k and the repetition times as constants:

(i) *the constrained k-means* [39] with Xing et al.'s metric learning algorithm (*CopK_Xing*);

(ii) *pairwise constrained k-means* with Xing et al.'s metric learning algorithm (*PCK_Xing*) [40, 41].

For Xing et al.'s metric learning method, the code is downloaded from Xing's home page: http://www.cs.cmu.edu/~epxing/publications.html.

We conduct experiments on twenty UCI real world datasets obtained from the Machine Learning Repository of the University of California, Irvine [42]. The information about those datasets is summarized in Table 1.

Table 1: The information of benchmark datasets.

Dataset	Abbr.	#Class	#Attr	Size
Balance	Bal	3	4	625
Breast Cancer Wisconsin	BCW	2	9	699
Credit	Cre	2	15	653
Ecoli	Eco	8	7	336
Hepatitis	Hep	2	19	155
Housing	Hou	3	13	506
Ionosphere	Ion	2	34	351
Iris	Iri	3	4	150
mfeat-fac	Mff	10	216	2000
mfeat-pix	Mfp	10	240	2000
Pima	Pim	2	8	768
Promoters	Pro	2	57	106
Segmentation	Seg	7	19	2310
Sick	Sic	2	29	3772
Soybean	Soy	4	35	47
Splice	Spl	3	60	3175
Voting	Vot	2	16	435
Wine	Win	3	13	178
Yeast	Yea	10	8	1484
Zoo	Zoo	7	17	101

5.2. The Experiments Setup. We first make the following preprocessing: for a nominal attribute with I different values, we replace these values by I integers $1, 2, \ldots, I$, and then all attributes are normalized to the interval $[1, 2]$.

Except *Ecoli*, $|S_i|$ is set to five for $i = 1, 2, \ldots, k$. Because the smallest number of instances is two among eight classes in the dataset *Ecoli*, $|S_i|$ is set to two for $i = 1, 2, \ldots, k$.

Table 2: The mean Rand Index and the standard deviation over 20 random runs ($|S_i| = 2$ for *Ecoli* and 5 for the others, $i = 1, 2, \ldots, k$).

Dataset	FPC	NNC	FPC_Diag	NNC_Diag	CopK_Xing	PCK_Xing	FPC_Xing	NNC_Xing
Bal	0.534 ± 0.034	0.594 ± 0.038	0.510 ± 0.033	0.588 ± 0.032	0.602 ± 0.039	0.594 ± 0.031	0.434 ± 0.002	**0.608** ± 0.067
BCW	0.629 ± 0.074	0.832 ± 0.044	0.636 ± 0.050	0.803 ± 0.023	0.840 ± 0.150	**0.860** ± 0.135	0.579 ± 0.014	0.846 ± 0.032
Cre	0.521 ± 0.013	0.601 ± 0.066	0.538 ± 0.050	0.660 ± 0.077	**0.683** ± 0.046	**0.685** ± 0.065	0.506 ± 0.006	0.625 ± 0.075
Eco	0.596 ± 0.089	**0.871** ± 0.022	0.716 ± 0.093	0.793 ± 0.021	0.816 ± 0.009	0.818 ± 0.011	0.298 ± 0.013	0.813 ± 0.033
Hep	0.599 ± 0.063	0.579 ± 0.084	0.640 ± 0.039	0.572 ± 0.063	0.566 ± 0.036	0.564 ± 0.039	**0.668** ± 0.010	0.588 ± 0.066
Hou	0.546 ± 0.023	0.603 ± 0.028	0.497 ± 0.048	**0.621** ± 0.024	0.604 ± 0.005	0.601 ± 0.006	0.467 ± 0.024	0.607 ± 0.040
Ion	0.519 ± 0.031	0.552 ± 0.023	0.549 ± 0.034	0.553 ± 0.036	0.571 ± 0.021	**0.581** ± 0.013	0.522 ± 0.013	0.543 ± 0.021
Iri	0.618 ± 0.056	0.870 ± 0.021	0.655 ± 0.025	0.907 ± 0.068	0.845 ± 0.061	0.819 ± 0.129	0.446 ± 0.096	**0.918** ± 0.026
Mff	0.691 ± 0.040	0.879 ± 0.012	0.787 ± 0.047	**0.915** ± 0.013	0.901 ± 0.014	0.903 ± 0.008	0.733 ± 0.019	0.875 ± 0.009
Mfp	0.377 ± 0.084	0.867 ± 0.013	0.795 ± 0.053	**0.906** ± 0.016	**0.906** ± 0.016	**0.909** ± 0.015	0.730 ± 0.018	0.880 ± 0.014
Pim	0.542 ± 0.014	0.544 ± 0.027	0.540 ± 0.018	0.538 ± 0.034	**0.556** ± 0.004	**0.555** ± 0.004	0.544 ± 0.001	**0.553** ± 0.024
Pro	0.502 ± 0.008	0.537 ± 0.031	0.511 ± 0.025	0.572 ± 0.061	**0.588** ± 0.063	0.579 ± 0.072	0.497 ± 0.003	0.574 ± 0.033
Seg	0.589 ± 0.103	**0.855** ± 0.015	0.397 ± 0.162	**0.854** ± 0.049	0.827 ± 0.038	0.843 ± 0.021	0.403 ± 0.099	0.821 ± 0.016
Sic	0.595 ± 0.083	0.631 ± 0.124	0.789 ± 0.114	0.835 ± 0.151	0.679 ± 0.136	0.658 ± 0.142	**0.863** ± 0.018	0.652 ± 0.095
Soy	0.669 ± 0.044	0.973 ± 0.022	0.754 ± 0.049	**0.982** ± 0.011	0.906 ± 0.080	0.843 ± 0.087	0.715 ± 0.056	0.950 ± 0.014
Spl	0.528 ± 0.005	0.516 ± 0.028	0.510 ± 0.031	0.547 ± 0.018	**0.618** ± 0.039	**0.619** ± 0.036	0.385 ± 0.000	0.539 ± 0.026
Vot	0.540 ± 0.036	0.612 ± 0.097	0.573 ± 0.074	0.769 ± 0.124	**0.773** ± 0.004	0.712 ± 0.111	0.526 ± 0.019	0.731 ± 0.071
Win	0.607 ± 0.038	0.804 ± 0.048	0.567 ± 0.076	**0.883** ± 0.038	0.840 ± 0.085	0.807 ± 0.088	0.363 ± 0.008	0.803 ± 0.032
Yea	0.291 ± 0.055	0.681 ± 0.031	0.461 ± 0.028	0.679 ± 0.034	**0.723** ± 0.012	**0.726** ± 0.013	0.233 ± 0.002	0.644 ± 0.049
Zoo	0.807 ± 0.041	0.983 ± 0.012	0.849 ± 0.047	**0.992** ± 0.016	0.928 ± 0.046	0.914 ± 0.039	0.753 ± 0.122	0.981 ± 0.010
Mean	0.565 ± 0.047	0.719 ± 0.039	0.614 ± 0.055	**0.748** ± 0.046	0.739 ± 0.046	0.730 ± 0.053	0.533 ± 0.027	0.728 ± 0.038

Xing et al.'s metric learning is carried out on the original pairwise constraints: $\mathrm{ML} = \{(p, q) \mid p, q \in S_i, i = 1, 2, \ldots, k\}$ and $\mathrm{CL} = \{(p, q) \mid p \in S_i, q \in S_j, i, j = 1, 2, \ldots, k, i \neq j\}$. In the phase of clustering for *CopK_Xing* and *PCK_Xing*, it is the centroid c_i of S_i that participates in the clustering process, which guarantees that all must-link constraints are satisfied.

The stop condition is either the repetition times are more than 100 or the objective difference between two consecutive repetitions is less than 10^{-6}.

We use the Rand Index [43] to measure the clustering quality in our experiments. The Rand Index reflects the agreement of the clustering result with the ground truth. Here, the ground truth is given by the data's class labels. Let n_s be the number of instance pairs that are assigned to the same cluster and have the same class label, and let n_d be the number of instance pairs that are assigned to different clusters and have different class labels. Then, the Rand Index is defined as

$$\mathrm{RI} = \frac{2\left(n_s + n_d\right)}{\left(n\left(n-1\right)\right)}. \tag{27}$$

All algorithms are implemented in MATLAB R2009b, and experiments are carried out on a 2.6 GHz double-core Pentium PC with 2 G bytes of RAM.

5.3. The Mean Rand Index. Table 2 summarizes the mean Rand Index and the standard deviation over 20 random runs on twenty datasets, and the value with bold in each row is the highest. Table 2 shows that although no algorithm performs better than the other algorithms on all datasets, in general we can draw the following conclusion.

(1) The supervision can significantly improve the clustering quality: compared with *FPC* and *FPC_Diag*, the mean Rand Index of *NNC* and *NNC_Diag* over twenty datasets increases about 27 percent and 22 percent, respectively. Note that the increment of the Rand Index that resulted from the addition of supervision itself is very small.

(2) The introducing of metric learning into an existing algorithm does not always increase its performance. However, in general, the effect of our metric learning model is positive: the win/loss ratio of *FPC_Diag* to *FPC* is 11/3, and the win/loss ratio of *NNC_Diag* to *NNC* is 8/1, where *algorithm A* defeating *algorithm B* means that the Rand Index of *A* is higher at least 0.03 than that of *B* since the standard deviation is a bit large.

(3) Compared with *CopK_Xing* and *PCK_Xing*, *NNC_Diag* performs a little better: the win/loss ratio of *NNC_Diag* to *CopK_Xing* is 5/3, and the win/loss ratio of *NNC_Diag* to *PCK_Xing* is 6/3.

(4) For the *FPC* and *NNC* clustering algorithms, the proposed metric learning model is better than Xing et al.'s method, especially for *FPC*. For *FPC*, Xing et al.'s method resulted in the fact that the performance of *FPC* significantly decreased on nine datasets and the mean Rand Index of *FPC_Xing* even decreases about 6 percent compared with *FPC*. The win/loss ratio of *NNC_Diag* to *NNC_Xing* is 8/1. This fact seems to

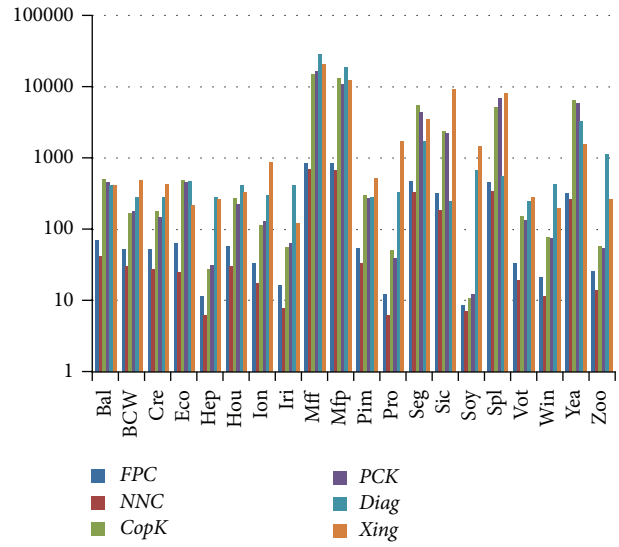

Figure 1: The logarithm graph of the mean runtime (milliseconds) over 20 random runs.

advise that when selecting a metric learning model for an existing clustering algorithm, the metric learning model should correspond to the clustering criterion of the clustering algorithm.

5.4. The Runtime. Figure 1 depicts the logarithm graph of the mean runtime (milliseconds) over 20 random runs, where the runtime of *FPC*, *NNC*, *CopK*, and *PCK* does not include the metric learning time. The legend *Diag* denotes the runtime of the metric learning time of our diagonal matrix model, and the legend *Xing* denotes the metric learning time of Xing et al.'s model (the diagonal matrix). So, the runtime of *FPC_Diag* (*NNC_Diag*) is the sum of the *FPC (NNC)* and the *Diag*. Similarly, the runtime of *CopK_Xing* (*PCK_Xing*) is the sum of the *CopK* (*PCK*) and the *Xing*.

Figure 1 shows that both *NNC* and *FPC* are much faster than *CopK* and *PCK*, which is consistent with their time complexities: the complexity of *FPC* and *NNC* is $O(nk)$, whereas the complexity of *CopK* and *PCK* is $O(nkt)$, where t is repetition times of *k-means*. Figure 1 also shows that Xing et al.'s model is slower than our model when the number of dimensions is relatively large, for example, *Ionosphere*, *Promoters*, *Sick*, and *Splice*. On the other hand, since the number of inequality constraints is quadratic with the number of class labels, our *Diag* model is slower than Xing et al.'s model on datasets with relatively large number of class labels, for example, *Ecoli*, *Mfeat-fac*, *Mfeat-pix*, *Yeast*, and *Zoo*.

The experimental results in Table 2 and Figure 1 show that the *FPC* algorithm is very fast, but the clustering results are unsatisfactory. The *NNC* algorithm proposed in this paper has the same time complexity as *FPC*, but the clustering quality is much more satisfactory than *FPC* if a few labeled instances are available.

6. Conclusion

In this paper, we studied the problem related to clusterability. We showed that if the input data are well clusterable, the optimal solutions with respect to the min-max diameter criterion, the max-min split criterion, and the max-RSD criterion can be simultaneously found in linear time for both unsupervised and semisupervised learning. For the max-RSD criterion, we also proposed two convex optimization models to make data more clusterable.

The experimental results on twenty UCI datasets demonstrate that both the supervision and the learned metric can significantly improve the clustering quality. We believe that the proposed *NNC* algorithm and metric learning models are useful when only a few labeled instances are available.

Usually, the term semisupervised learning is used to describe scenarios where both the labeled data and the unlabeled date affect the performance of a learning algorithm, which is not the case here: the supervised data is used either to induce a nearest neighbor classifier on the unlabeled data or to find a metric vector. Hence, the supervision information can be more elaborately utilized in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by China Natural Science Foundation under Grant no. 61273363 and Natural Science Foundation of Guangdong Province under Grant no. 06300170.

References

[1] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, no. 2-3, pp. 293–306, 1985.

[2] D. S. Hochbaum and D. B. Shmoys, "A unified approach to approximation algorithms for bottleneck problems," *Journal of the ACM*, vol. 33, no. 3, pp. 533–550, 1986.

[3] T. Feder and D. H. Greene, "Optimal algorithms for approximate clustering," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC '88)*, pp. 434–444, ACM, May 1988.

[4] S. Dasgupta and P. M. Long, "Performance guarantees for hierarchical clustering," *Journal of Computer and System Sciences*, vol. 70, no. 4, pp. 555–569, 2005.

[5] P. Hansen and B. Jaumard, "Minimum sum of diameters clustering," *Journal of Classification*, vol. 4, no. 2, pp. 215–226, 1987.

[6] M. Charikar and R. Panigrahy, "Clustering to minimize the sum of cluster diameters," *Journal of Computer and System Sciences*, vol. 68, no. 2, pp. 417–441, 2004.

[7] S. Ramnath, "Dynamic digraph connectivity hastens minimum sum-of-diameters clustering," *SIAM Journal on Discrete Mathematics*, vol. 18, no. 2, pp. 272–286, 2004.

[8] M. Gibson, G. Kanade, E. Krohn, I. A. Pirwani, and K. Varadarajan, "On metric clustering to minimize the sum of radii," *Algorithmica*, vol. 57, no. 3, pp. 484–498, 2010.

[9] J. Wang and J. Chen, "Clustering to maximize the ratio of split to diameter," in *Proceedings of the 29th International Conference on Machine Learning (ICML '12)*, pp. 241–248, Edinburgh, Scotland, June-July 2012.

[10] M. Ackerman and S. Ben-David, "Measures of clustering quality: a working set of axioms for clustering," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*, pp. 121–128, Vancouver, Canada, December 2008.

[11] M. Ackerman and S. Ben-David, "Clusterability: a theoretical study," in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, vol. 5 of *JMLR: W&CP 5*, pp. 1–8, 2009.

[12] O. Chapelle, B. Schökopf, and A. Zien, Eds., *Semi-Supervised Learning*, MIT Press, Cambridge, Mass, USA, 2006.

[13] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2008, http://pages.cs.wisc.edu/~jerryzhu/research/ssl/semireview.html.

[14] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," in *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pp. 1103–1110, Morgan Kaufmann, Palo Alto, Calif, USA, 2000.

[15] I. Davidson and S. Basu, "A survey of clustering with instance-level constraints," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, pp. 1–41, 2007.

[16] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 607–616, 1996.

[17] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proceedings of the 20th International Conference on Machine Learning*, pp. 11–18, August 2003.

[18] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 15, pp. 505–512, MIT Press, 2003.

[19] H. Chang and D.-Y. Yeung, "Locally linear metric adaptation for semi-supervised clustering," in *Proceedings of the 21 International Conference on Machine Learning (ICML '04)*, pp. 153–160, July 2004.

[20] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '05)*, 2005.

[21] S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 2072–2078, June 2006.

[22] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 905–912, June 2006.

[23] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2006.

[24] L. Yang, R. Jin, R. Sukthankar, and Y. Liu, "An efficient algorithm for local distance metric learning," in *Proceedings of the 21st national conference on Artificial intelligence (AAAI '06)*, pp. 543–548, July 2006.

[25] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 209–216, ACM, Corvallis, Ore, USA, June 2007.

[26] S. C. H. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–7, June 2008.

[27] K. Huang, Y. Ying, and C. Campbell, "GSML: a unified framework for sparse metric learning," in *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM '09)*, pp. 189–198, December 2009.

[28] G.-J. Qi, J. Tang, Z.-J. Zha, T.-S. Chua, and H.-J. Zhang, "An efficient sparse metric learning in high-dimensional space via L_1-penalized log-determinant regularization," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pp. 841–848, 2009.

[29] Y. Xu, W. Ping, and A. T. Campbell, "Multi-instance metric learning," in *Proceedings of the 9th IEEE International Conference on Data Mining*, pp. 874–883, December 2009.

[30] R. Jin, S. Wang, and Y. Zhou, "Regularized distance metric learning: theory and algorithm," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2010.

[31] W. Liu, X. Tian, D. Tao, and J. Liu, "Constrained metric learning via distance gap maximization," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence Conference*, pp. 518–524, Atlanta, Ga, USA, July 2010.

[32] Y. Hong, Q. Li, J. Jiang, and Z. Tu, "Learning a mixture of sparse distance metrics for classification and dimensionality reduction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 906–913, November 2011.

[33] J. Wang, H. Do, A. Woznica, and A. Kalousis, "Metric learning with multiple kernels," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '11)*, 2011.

[34] E. Y. Liu, Z. Guo, X. Zhang, V. Jojic, and W. Wang, "Metric learning from relative comparisons by minimizing squared residual," in *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM '12)*, pp. 978–983, December 2012.

[35] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *Journal of Machine Learning Research*, vol. 13, pp. 1–26, 2012.

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.

[37] I. Davidson and S. S. Ravi, "Clustering with constraints: feasibility issues and the k-means algorithm," in *Proceedings of the 5th SIAM International Conference on Data Mining (SDM '05)*, pp. 138–149, April 2005.

[38] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, Calif, USA, 1979.

[39] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *Proceedings of the 18th International Conference on Machine Learning*, pp. 577–584, 2001.

[40] S. Basu, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Proceedings of the SIAM International Conference on Data Mining*, pp. 333–344, Lake Buena Vista, Fla, USA, 2004.

[41] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of the 21st International Conference on Machine Learning*, pp. 81–88, Banff, Canada, July 2004.

[42] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, School of Information and Computer Science, University of California, Irvine, Calif, USA, 2010, http://archive.ics.uci.edu/ml.

[43] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.