Semi-Supervised Maximum Margin Clustering with Pairwise Constraints

Hong Zeng, *Member*, *IEEE*, and Yiu-Ming Cheung, *Senior Member*, *IEEE*

Abstract—The pairwise constraints specifying whether a pair of samples should be grouped together or not have been successfully incorporated into the conventional clustering methods such as k-means and spectral clustering for the performance enhancement. Nevertheless, the issue of pairwise constraints has not been well studied in the recently proposed *maximum margin clustering* (MMC), which extends the maximum margin framework in supervised learning for clustering and often shows a promising performance. This paper therefore proposes a pairwise constrained MMC algorithm. Based on the maximum margin idea in MMC, we propose a set of effective loss functions for discouraging the violation of given pairwise constraints. For the resulting optimization problem, we show that the original nonconvex problem in our approach can be decomposed into a sequence of convex quadratic program problems via constrained concave-convex procedure (CCCP). Subsequently, we present an efficient subgradient projection optimization method to solve each convex problem in the CCCP sequence. Experiments on a number of real-world data sets show that the proposed constrained MMC algorithm is scalable and outperforms the existing constrained MMC approach as well as the typical semi-supervised clustering counterparts.

Index Terms—Semi-supervised clustering, pairwise constraints, maximum margin clustering, constrained concave-convex procedure.

1 Introduction

Traditionally, unsupervised clustering algorithms have been widely used to discover the structure of grouping in the data. Recently, there is an emerging interest in incorporating limited supervision information into clustering algorithms to obtain user desired and more accurate partition. In this paper, we focus on the semi-supervised clustering (also called constrained clustering interchangeably), where the pairwise constraint provides the supervision information: a *must-link* (**ML**) constraint specifies that the pair of instances should be assigned to the same cluster, and a *cannot-link* (**CL**) constraint specifies that the pair of instances should be placed into the different clusters. From the practical viewpoint, the utilization of pairwise constraints is often a natural choice. In some application domains, the pairwise constraints can be collected automatically along with the unlabeled data. For example, the protein co-occurring information in the Database of Interacting Proteins (DIP) data set, can be used as the *must-link* constraints when performing gene clustering [1]. Furthermore, it is relatively easier for a user, who is even not an expert in a domain, to make a judgment whether two objects are similar or not than to provide them with the exact class labels.

- *H. Zeng is with the School of Instrument Science and Engineering, Southeast University, China, and the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China. E-mail: hzeng@seu.edu.cn.*
- *Y.-M. Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China. E-mail: ymc@comp.hkbu.edu.hk.*

Manuscript received 21 July 2009; revised 16 Feb. 2010; accepted 27 Sept. 2010; published online 7 Mar. 2011.
Recommended for acceptance by J. Li.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2009-07-0563.
Digital Object Identifier no. 10.1109/TKDE.2011.68.

Existing semi-supervised clustering can be generally classified into two lines. In the first line, the constraints are used to learn a Mahalanobis distance measure [3], [4], [5]. Then, a traditional clustering algorithm, e.g., K-means, is performed under this new distance measure. Although the improvement over completely unsupervised clustering has often been achieved, it is generally known that such a method requires a large number of pairwise constraints to obtain a reliable estimation of parameters in a distance metric [6]. In the second line, there are two ways to directly enforce the constraints in a specific clustering algorithm. One way is to strictly require that any of the constraints should not be violated during the clustering process [7]. The other way is in a more appropriate manner. It augments the traditional clustering objective functions with the penalty terms for violating the constraints [6], [8], [9], [10], [1], [11], which is more robust against "noisy" pairwise constraints. In the literature, the constrained K-means [6], [8], [12], constrained Gaussian mixtures [9], [10], and constrained spectral clustering [1], [11] have all been developed along this line [2].

Along the second line, this paper is interested in incorporating the pairwise constraints into the recently proposed *maximum margin clustering* (MMC) [13], [14], [15], [16], [17]. MMC utilizes the maximum margin principle adopted in the supervised learning and tries to find the hyperplanes that partition the data into different clusters with the largest margins between them over all the possible labelings. Recent studies [13], [14], [15], [16], [17] have shown the promising performance of MMC. Nevertheless, the accuracy of the clustering results by MMC may not be satisfactory sometimes due to the nature of its unsupervised learning. In [18], a preliminary study on constrained MMC shows that incorporating the pairwise constraints can improve the performance of the basic MMC. Despite the

success in its application domain, the loss function for violating the *cannot-link* constraints in [18] may not be able to discourage the violation of given constraints robustly, especially for the data sets whose samples from different categories are *heavily* overlapped, e.g., the images with similar appearance but belong to different ground-truth classes, the documents with many words in common but talk about different topics, and so on. Further, it is generally known that the *must-link* constraints are less informative in forming the partitioning boundary [19] than the *cannot-link* ones. Hence, the improvement by Hu et al. [18] may not be effective in such a case.

To this end, we propose a new semi-supervised maximum margin clustering algorithm in this paper. The main contributions of our work are two-fold: 1) We introduce a new set of loss functions, featuring the robust performance of penalizing the violation of given pairwise constraints under the maximum margin principle. 2) For the resulting optimization problem, although it is nonconvex, we show that the optimization can be carried out iteratively by solving a sequence of convex quadratic problems via the *constrained concave-convex procedure* (CCCP). Subsequently, we present a simple, effective, and fast iterative procedure for solving each convex problem in the CCCP sequence, which alternates between a subgradient descent step and a projection step. Experimental results show that the proposed constrained MMC algorithm is efficient, scalable, and outperforms the existing constrained MMC in [18] as well as some typical semi-supervised clustering counterparts.

The remainder of the paper is organized as follows: In Section 2, we briefly review the maximum margin clustering, and the preliminary constrained MMC algorithm developed in [18], as well as the other related work. The proposed constrained MMC is presented in Section 3. Experimental results are given in Section 4. Finally, we draw a conclusion in Section 5.

2 Related Work

Given n data points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ ($\mathbf{x}_i \in \mathbb{R}^d$), we group the data into C clusters and obtain a labeling vector $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$, where $\mathcal{Y} = \{1, \ldots, C\}$ and C is the number of clusters. First, a joint feature representation $\Phi(\mathbf{x}, y)$ for each $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is defined by

$$\Phi(\mathbf{x}, y) = \begin{bmatrix} \mathbf{x} \cdot I(y = 1) \\ \cdots \\ \mathbf{x} \cdot I(y = C) \end{bmatrix},$$

where $I(\cdot)$ is the indicator function (1 for "true" and 0 otherwise). MMC aims to find the maximum margin hyperplanes that can partition the data into different clusters over all possible labelings [13], [14], [15], [16], [17]. Suppose the hyperplanes are parameterized by a weight vector $\mathbf{W} \in \mathbb{R}^{(d \times C) \times 1}$, the multiclass MMC is then presented as the following optimization problem [16], which is based on the multiclass SVM formulation in [20]

$$\begin{aligned} &\min_{\mathbf{W}, \boldsymbol{\xi}} \frac{\lambda}{2}\|\mathbf{W}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ &s.t.\ \forall i, \forall s_i, z_i \in \mathcal{Y}, \\ &\quad \max_{y_i \in \mathcal{Y}} \mathbf{W}^T\Phi(\mathbf{x}_i, y_i) - \mathbf{W}^T\Phi(\mathbf{x}_i, z_i) \\ &\qquad + I\Big(\max_{y_i \in \mathcal{Y}} \mathbf{W}^T\Phi(\mathbf{x}_i, y_i) = \mathbf{W}^T\Phi(\mathbf{x}_i, z_i)\Big) \geq 1 - \xi_i, \\ &\quad -q \leq \sum_{i=1}^{n} \mathbf{W}^T\Phi(\mathbf{x}_i, s_i) - \sum_{i=1}^{n} \mathbf{W}^T\Phi(\mathbf{x}_i, z_i) \leq q, \\ &\qquad \xi_i \geq 0, \end{aligned} \tag{1}$$

where λ is a positive regularization constant, $\|\cdot\|$ is the l_2 norm, and T denotes the transpose operation. The value $\mathbf{W}^T\Phi(\mathbf{x}, y)$ is the score for sample $\mathbf{x}$ associated with the cluster y. The first inequity in (1) essentially requires that the score for assigning a sample to some cluster to which it is most likely to belong should be greater than the scores for assigning it to any other clusters by at least a margin of $(1 - \xi_i)$, where $\xi_i (i = 1, \ldots, n)$ is a slack variable. In this manner, the samples are enforced to be far away from these hyperplanes. That is, such a clustering scheme favors a low-density separation. The second inequity in (1) is a cluster balance constraint introduced to avoid the "trivially" optimal solutions because a large margin value can be always achieved by eliminating classes [21]. The constant $q \geq 0$ controls the cluster imbalance. Ultimately, MMC learns a hypothesis: $h : \mathcal{X} \rightarrow \mathcal{Y}$ via solving $\mathbf{W}$ and $\boldsymbol{\xi}$ in (1), and finally the cluster label for $\mathbf{x}_i$ is obtained by $\hat{y}_i^* = h(\mathbf{x}_i) = \arg\max_{y \in \mathcal{Y}} \widehat{\mathbf{W}}^T\Phi(\mathbf{x}_i, y)$.

Now a set of pairwise constraints $\{(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j)\}_{j=1}^L$ is given, where $l_j = 1$ indicates that the pair of samples $(\mathbf{x}_{j1}, \mathbf{x}_{j2})$ *must link*, while $l_j = -1$ means that the pair of samples $(\mathbf{x}_{j1}, \mathbf{x}_{j2})$ *cannot link*. A preliminary work on constrained MMC has been proposed in [18]. It is based on the work [22] that incorporates the pairwise constraints for classification. The objective function of [18] is presented below

$$\begin{aligned} &\min_{\mathbf{W}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\eta}} \frac{1}{2}\|\mathbf{W}\|^2 + \frac{\gamma_1}{nC}\sum_{i, z_i}\xi_{iz_i} \\ &\qquad + \frac{\gamma_2}{L_m C}\sum_{z_j, j: l_j = 1}\zeta_{jz_j} + \frac{\gamma_3}{L_c}\sum_{j: l_j = -1}\eta_j \\ &s.t.\ \forall i, \forall z_i \in \mathcal{Y}: \\ &\quad \max_{y_i \in \mathcal{Y}} \mathbf{W}^T\Phi(\mathbf{x}_i, y_i) - \mathbf{W}^T\Phi(\mathbf{x}_i, z_i) \geq 1 - \xi_{iz_i}, \\ &\quad \xi_{iz_i} \geq 0, \\ &\forall j, \forall z_j \in \mathcal{Y}, l_j = 1\ (\mathbf{ML}): \\ &\quad \zeta_{jz_j} \geq \big|\mathbf{W}^T\Phi(\mathbf{x}_{j1}, z_j) - \mathbf{W}^T\Phi(\mathbf{x}_{j2}, z_j)\big|, \\ &\forall j, \forall z_j \in \mathcal{Y}, l_j = -1\ (\mathbf{CL}): \\ &\quad \eta_j \geq \mathbf{W}^T\big[\Phi(\mathbf{x}_{j1}, z_j) + \Phi(\mathbf{x}_{j2}, z_j)\big] \\ &\qquad - \frac{1}{C}\sum_{s=1}^{C}\mathbf{W}^T\big[\Phi(\mathbf{x}_{j1}, s) + \Phi(\mathbf{x}_{j2}, s)\big], \end{aligned} \tag{2}$$

where L_m and L_c are the numbers of *must-link* constraints and *cannot-link* constraints, respectively. γ_1, γ_2, and γ_3 are positive constants that balance the l_2 norm regularizer and

the loss functions. For *must-link* constraints, it tries to find a solution that leads to a small difference between the scores for the pair of instances associated with the same cluster. The *cannot-link* constraints in (2) essentially require that none of the clusters should declare the ownership for the *cannot-link* pair of samples. Specifically, it minimizes $\sum_j \eta_j$ so as to discourage the unwanted case, where the sum of scores for $(\mathbf{x}_{j1}, \mathbf{x}_{j2}, \forall l_j = -1)$ associated with some cluster is significantly greater than the average level for all the clusters. The balance constraints are not imposed in (2) because the *cannot-link* constraints are able to prevent the "trivially" optimal solutions [18].

In (2), it can be seen that the variable $\eta_{j:l_j=-1}$ should be determined by such a cluster that has the largest score for possessing both instances in the *cannot-link* pair among all the clusters, i.e.,

$$\eta_{j:l_j=-1} = \max_{z_j \in \mathcal{Y}} \mathbf{W}^T \big[\Phi(\mathbf{x}_{j1}, z_j) + \Phi(\mathbf{x}_{j2}, , z_j)\big] - \frac{1}{C}\sum_{s=1}^{C} \mathbf{W}^T \big[\Phi(\mathbf{x}_{j1}, s) + \Phi(\mathbf{x}_{j2}, s)\big]. \tag{3}$$

In fact, $\eta_{j:l_j=-1}$ is just the value of the loss function for violating the *cannot-link* constraint $(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j = -1)$. We now investigate its robustness in two cases shown in Fig. 1, i.e., a relatively easy one and a more difficult one. For the relatively easy case in Fig. 1a, where the samples from different categories are *slightly* overlapped (i.e., the current premature solution $\mathbf{W}$ satisfies $\max_{y_i \in \mathcal{Y}} \mathbf{W}^T \Phi(\mathbf{x}_i, y_i) - \mathbf{W}^T \Phi(\mathbf{x}_i, z_i) \geq 1$ for most samples) and $\mathbf{W}$ has determined that the cluster 1 superiorly owns the *cannot-link* pair of samples, the loss then will be a large positive value according to (3), which can therefore penalize such a violation of the given constraint. Nevertheless, the *cannot-link* constraints usually attempt to guide the partitioning in the more difficult scenarios where the samples from different classes are seriously overlapped. As shown in Fig. 1b, a *cannot-link* constraint has been provided in the *severely* overlapping region, in which $\mathbf{x}_{j1}$ and $\mathbf{x}_{j2}$ belong to different ground-truth categories but are very similar. Moreover, in Fig. 1b, none of the three clusters have the absolute superior ownership of advantage over the others for $\mathbf{x}_{j1}$ nor $\mathbf{x}_{j2}$. In other words, the current premature solution $\mathbf{W}$ heavily violates the margin requirement specified in the first inequity of (2), i.e., $\max_{y_i \in \mathcal{Y}} \mathbf{W}^T \Phi(\mathbf{x}_i, y_i)$ is close to $\mathbf{W}^T \Phi(\mathbf{x}_i, z_i), \forall z_i \in \mathcal{Y}$ for samples in this region. Subsequently, such the largest score, i.e., $\max_{z_j \in \mathcal{Y}} \mathbf{W}^T [\Phi(\mathbf{x}_{j1}, z_j) + \Phi(\mathbf{x}_{j2}, z_j)]$, may be just slightly greater than the average one: $\frac{1}{C}\sum_{s=1}^{C} \mathbf{W}^T [\Phi(\mathbf{x}_{j1}, s) + \Phi(\mathbf{x}_{j2}, s)]$, resulting in $\eta_{j:l_j=-1}$ in (3) very close to 0 as if a *cannot-link* constraint were not been violated. That is, in such a difficult case, even if the current solution $\mathbf{W}$ has failed to place the samples from the *cannot-link* constraint into two well-separated clusters, the loss for such a violation will be instead close to zero, being unable to effectively discourage the premature solution. Hence, we will present a new set of loss functions for pairwise constraints to overcome this possible drawback.

Our work can be considered as an extension to the works in [22], [23]. They address the semi-supervised

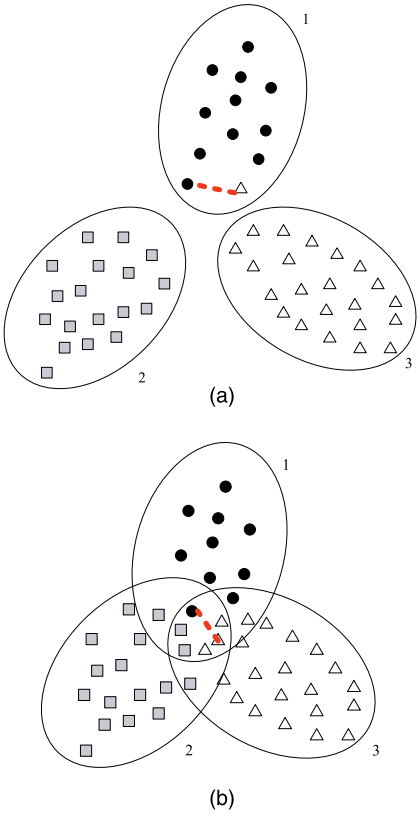


Fig. 1. The premature clustering results in two cases of different overlapping levels. The circles, squares, and triangles represent the samples from different categories. The ellipses denote the clusters determined by a premature solution $\mathbf{W}$. The dashed line denotes the *cannot-link* pairwise constraint $(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j = -1)$. (a) In this case, the loss for violating the *cannot-link* constraint in [18] will be a large positive value, providing effective penalization to $\mathbf{W}$ which results in the *cannot-link* pair of samples being partitioned into the same group, i.e., Cluster 1. (b) In this case, the loss for violating the *cannot-link* constraint in [18] will be close to zero, which is unable to effectively discourage the premature solution that fails to satisfy the given *cannot-link* constraint. (a) The slightly overlapping case. (b) The severely overlapping case.

classification problems where a small number of labels for the training set together with some additional pairwise constraints are known. Promising performance has been reported in both papers [22], [23]. Nevertheless, the unlabeled samples, which may provide useful information for the semi-supervised learning, are completely ignored. Moreover, Nguyen and Caruana [23] directly utilizes a solver, which is designed for convex programming, to a nonconvex optimization problem. Besides, the most recently proposed MMC [16] also has a similar deficiency in optimizing the problem of (1). It directly solves a nonconvex problem using the cutting plane method [24] which is also designed for solving convex problems. Consequently, the optimality and convergence of the algorithms in [23], [16] may not be guaranteed. By contrast, we utilize CCCP to decompose the original nonconvex problem into a series of convex quadratic program problems, and then find solutions to such convex subproblems with a convex solver. In this way, the proposed algorithm can be guaranteed to converge to a local optimal solution. Hence, the proposed optimization method is more theoretically sound than the methods in [23], [16].

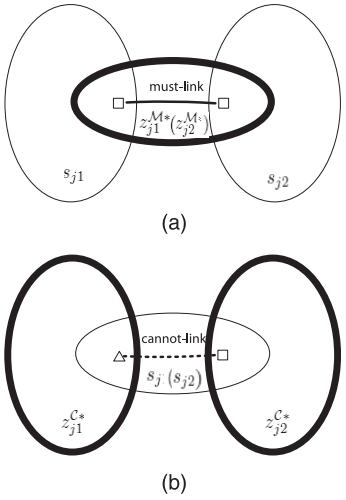


Fig. 2. The most desired clusters and undesired ones for the pairwise constraints. The triangle and square denote the samples from different categories. The solid line and the dashed line between the samples represent the *must-link* constraint and the *cannot-link* constraint, respectively. The most desired clusters conforming to the given constraints are marked by the ellipses with the thick lines, whereas the undesired clusters violating the given constraints are marked by the ellipses with the thin lines. (a) The *must-link* constraint. (b) The *cannot-link* constraint.

3 The Proposed Pairwise Constrained Maximum Margin Clustering

In order to estimate the weight vector $\mathbf{W}$ robustly, we will exploit the "margin" idea adopted in MMC to deal with the pairwise constraints.

Hereinafter, we will use the shorthand $\Phi(\mathbf{x}_i, \mathbf{x}_j, z_i, z_j) \equiv \Phi(\mathbf{x}_i, z_i) + \Phi(\mathbf{x}_j, z_j)$. We define the score that a pair of instances are assigned to the same cluster by the value: $\mathbf{W}^T\Phi(\mathbf{x}_i, \mathbf{x}_j, z_i, z_j)$, where $z_i = z_j$ and $z_i, z_j \in \mathcal{Y}$. Similarly, the score that a pair of instances are assigned to different clusters is defined by the value: $\mathbf{W}^T\Phi(\mathbf{x}_i, \mathbf{x}_j, z_i, z_j)$, where $z_i \neq z_j$ and $z_i, z_j \in \mathcal{Y}$. The main idea of our approach is that the score for the most possible assigning scheme satisfying the specified pairwise constraint should be always greater than that for any assigning scheme which violates the constraint by at least a margin.

Specifically, given a *must-link* constraint $(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j = 1)$, we require that the largest score for assigning $(\mathbf{x}_{j1}, \mathbf{x}_{j2})$ into the same cluster should be greater than that for assigning them into two different clusters by at least 1. Furthermore, we also allow the margin to be violated by η_j $(\eta_j \geq 0)$ at most for better estimation of $\mathbf{W}$. Namely, the following constraint is imposed on the vector $\mathbf{W}$

$$\begin{aligned}&\max_{z_{j1}=z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2}) - \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\\ &\quad \geq 1 - \eta_j, s_{j1}, s_{j2} \in \mathcal{Y},\ s_{j1} \neq s_{j2},\ \eta_j \geq 0,\end{aligned} \tag{4}$$

and $(z_{j1}^{\mathcal{M}*}, z_{j2}^{\mathcal{M}*}) = \arg\max_{z_{j1}=z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_j, z_{j1}, z_{j2})$ serves as the most desired assigning scheme conforming to the *must-link* constraint.

Analogously, given a *cannot-link* constraint $(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j = -1)$, we require that the largest score for assigning $(\mathbf{x}_{j1}, \mathbf{x}_{j2})$ into two different clusters should be greater than that for assigning them into the same cluster by at least $(1 - \eta_j)$

$$\begin{aligned}&\max_{z_{j1}\neq z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2}) - \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\\ &\quad \geq 1 - \eta_j, s_{j1}, s_{j2} \in \mathcal{Y}, s_{j1} = s_{j2}, \eta_j \geq 0,\end{aligned} \tag{5}$$

and $(z_{j1}^{\mathcal{C}*}, z_{j2}^{\mathcal{C}*}) = \arg\max_{z_{j1}\neq z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2})$ serves as the most desired assigning scheme complying with the *cannot-link* constraint. Our main idea for handling the given pairwise constraints is summarized in Fig. 2.

Ultimately, we obtain the following optimization problem for the constrained maximum margin clustering

$$\begin{aligned}&\min_{\mathbf{W},\boldsymbol{\eta},\boldsymbol{\xi}} \frac{\lambda}{2}\|\mathbf{W}\|^2 + \frac{1}{L}\sum_{j=1}^{L}\eta_j + \frac{\delta}{UC}\sum_{i\in\mathcal{U}}\sum_{z_i=1}^{C}\xi_{iz_i}\\ &s.t.\\ &\forall j, \forall s_{j1}, s_{j2} \in \mathcal{Y}, s_{j1} \neq s_{j2}, l_j = 1\ (\mathbf{ML}):\\ &\max_{z_{j1}=z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2}) - \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\\ &\quad \geq 1 - \eta_j, \quad \eta_j \geq 0,\\ &\forall j, \forall s_{j1}, s_{j2} \in \mathcal{Y}, s_{j1} = s_{j2}, l_j = -1\ (\mathbf{CL}):\\ &\max_{z_{j1}\neq z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2}) - \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\\ &\quad \geq 1 - \eta_j, \quad \eta_j \geq 0,\\ &\forall i \in \mathcal{U}, \quad \forall z_i \in \mathcal{Y},\\ &\max_{y_i\in\mathcal{Y}} \mathbf{W}^T\Phi(\mathbf{x}_i, y_i) - \mathbf{W}^T\Phi(\mathbf{x}_i, z_i) \geq 1 - \xi_{iz_i}, \quad \xi_{iz_i} \geq 0,\end{aligned} \tag{6}$$

where $\mathcal{U}$ denotes the set of indices for the unconstrained instances that do not involve in the constraint set, U is the size of $\mathcal{U}$, $L + U = n$, and $\delta \geq 0$ is a balancing constant.

For the *cannot-link* constraint in the optimization problem (6), the value of slack variable $\eta_{j:l_j=-1}$ should be determined by assigning $(\mathbf{x}_{j1}, \mathbf{x}_{j2})$ to some cluster that incurs the strongest margin violation, i.e.,

$$\begin{aligned}\eta_{j:l_j=-1} &= \max\left\{\max_{s_{j1}=s_{j2}}\left\{1 - \left[\max_{z_{j1}\neq z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2})\right.\right.\right.\\ &\quad \left.\left.\left. - \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\right]\right\}, 0\right\}\\ &= \max_{s_{j1}=s_{j2}}\left\{1 - \left[\max_{z_{j1}\neq z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2})\right.\right.\\ &\quad \left.\left. - \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\right]\right\}_+\\ &= \left\{1 - \left[\max_{z_{j1}\neq z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2})\right.\right.\\ &\quad \left.\left. - \max_{s_{j1}=s_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\right]\right\}_+,\end{aligned} \tag{7}$$

where $\{\theta\}_+ = \max\{\theta, 0\}$. Actually, $\eta_{j:l_j=-1}$ is the value of the loss function for $\mathbf{W}$ violating the *cannot-link* constraint $(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j = -1)$. For the case shown in Fig. 1b, although the samples $\mathbf{x}_{j1}$ and $\mathbf{x}_{j2}$ in the constraint are very similar and the current premature solution $\mathbf{W}$ makes the largest score for assigning them into the same cluster, i.e.,

$$\max_{s_{j1}=s_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2}),$$

very close to that for assigning them into two different clusters, i.e.,

$$\max_{z_{j1}\neq z_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1},z_{j2}),$$

the value of the loss function in (7), i.e., $\eta_{j:l_j=-1}$, will be steadily driven toward 1. In other words, when $\mathbf{W}$ has led to a partition that violates the *cannot-link* constraints provided in the severely overlapping region, the loss $\sum_j \eta_{j:l_j=-1}$ in (6) will increase dramatically, and the premature $\mathbf{W}$ will be therefore heavily penalized. As a result, it is expected that such a soft margin loss formulation for the *cannot-link* constraints is more robust than [18, (3)] in such a case.

3.1 The Constrained Concave-Convex Procedure for the Optimization

The objective function in (6) is convex, but the three sets of margin constraints are not convex. Nevertheless, these constraints are just the difference of convex functions. Therefore, the *constrained concave-convex procedure* can be utilized. It is an effective technique proposed recently to solve problems with concave-convex objective function under concave-convex constraints in the following form [25], [26], [27]

$$\begin{aligned}&\min_{\mathbf{x}} f_0(\mathbf{x}) - g_0(\mathbf{x})\\ &s.t.\ f_i(\mathbf{x}) - g_i(\mathbf{x}) \le c_i,\quad i=1,\ldots,p,\end{aligned} \tag{8}$$

where $f_i, g_i (i=0,\ldots,p)$ are convex and differentiable functions, and $c_i \in \mathbb{R}$. Given an initial guess on $\mathbf{x}_0$, in the $(t+1)$th iteration, CCCP first replaces $g_i(\mathbf{x})$ in (8) with its tangent at $\mathbf{x}_t$, and then solves the resulting convex problem for $\mathbf{x}_{t+1}$:

$$\begin{aligned}&\min_{\mathbf{x}} f_0(\mathbf{x}) - \left[g_0(\mathbf{x}_t) + \nabla g_0(\mathbf{x}_t)^T(\mathbf{x}-\mathbf{x}_t)\right]\\ &s.t.\ f_i(\mathbf{x}) - \left[g_i(\mathbf{x}_t) + \nabla g_i(\mathbf{x}_t)^T(\mathbf{x}-\mathbf{x}_t)\right] \le c_i,\\ &\quad i=1,\ldots,p,\end{aligned} \tag{9}$$

where $\nabla g_i(\mathbf{x}_t)(i=0,\ldots,p)$ is the gradient of $g_i(.)$ at $\mathbf{x}_t$. It can be shown that the objective (9) in each CCCP iteration decreases monotonically [26]. Furthermore, [26, Theorem 1] has shown that a saddle point in the Lagrange function corresponding to (8) is also a saddle point in the Lagrange function corresponding to (9) with the same set of dual variables upon the fact that the linearization of the nonconvex parts is tight at the saddle point of (9). Thereby, when CCCP converges, it actually arrives at a local minimum of (8) [26].

For our problem, note that $\{\max_{y_i\in\mathcal{Y}}\mathbf{W}^T\Phi(\mathbf{x}_i,y_i)\}$ in the third set of margin constraints of (6) is convex, but it is a nonsmooth function of $\mathbf{W}$. Therefore, we need to replace the gradient with the subgradient when computing the tangent [27]. The subgradient of it at $\mathbf{W}^{(t)}$ (i.e., the current estimation of $\mathbf{W}$ in the tth iteration of CCCP) is calculated as

$$\nabla\left[\max_{y_i\in\mathcal{Y}}\mathbf{W}^T\Phi(\mathbf{x}_i,y_i)\right]\Big|_{\mathbf{W}=\mathbf{W}^{(t)}} = \Phi(\mathbf{x}_i,y_i^{(t)}), \tag{10}$$

where

$$y_i^{(t)} = \arg\max_{y\in\mathcal{Y}} \mathbf{W}^{(t)T}\Phi(x_i,y). \tag{11}$$

Then, we can obtain the tangent of $\{\max_{y_i\in\mathcal{Y}}\mathbf{W}^T\Phi(\mathbf{x}_i,y_i)\}$, i.e., its first-order Taylor expansion at $\mathbf{W}^{(t)}$

$$\begin{aligned}&\max_{y_i\in\mathcal{Y}}\mathbf{W}^{(t)T}\Phi(\mathbf{x}_i,y_i) + \Phi(\mathbf{x}_i,y_i^{(t)})^T(\mathbf{W}-\mathbf{W}^{(t)})\\ &\quad=\mathbf{W}^T\Phi(\mathbf{x}_i,y_i^{(t)}).\end{aligned} \tag{12}$$

In a similar way, we can obtain the tangent of $\max_{z_{j1}=z_{j2}}\mathbf{W}^T\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1},z_{j2}), \forall l_j=1\ (\mathbf{ML})$ at $\mathbf{W}^{(t)}$

$$\mathbf{W}^T\Phi\left(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1}^{\mathcal{M}(t)},z_{j2}^{\mathcal{M}(t)}\right), \tag{13}$$

as well as the tangent of $\max_{z_{j1}\neq z_{j2}}\mathbf{W}^T\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1},z_{j2}), \forall l_j=-1\ (\mathbf{CL})$ at $\mathbf{W}^{(t)}$

$$\mathbf{W}^T\Phi\left(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1}^{\mathcal{C}(t)},z_{j2}^{\mathcal{C}(t)}\right), \tag{14}$$

where

$$\left(z_{j1}^{\mathcal{M}(t)},z_{j2}^{\mathcal{M}(t)}\right) = \arg\max_{z_{j1}=z_{j2}:l_j=1}\mathbf{W}^{(t)T}\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1},z_{j2}), \tag{15}$$

$$\left(z_{j1}^{\mathcal{C}(t)},z_{j2}^{\mathcal{C}(t)}\right) = \arg\max_{z_{j1}\neq z_{j2}:l_j=-1}\mathbf{W}^{(t)T}\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1},z_{j2}). \tag{16}$$

Following the CCCP, $\max_{y_i\in\mathcal{Y}}\mathbf{W}^T\Phi(\mathbf{x}_i,y_i)$, $\max_{z_{j1}=z_{j2}}\mathbf{W}^T\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1},z_{j2})$ and $\max_{z_{j1}\neq z_{j2}}\mathbf{W}^T\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1},z_{j2})$ in (6) are replaced with their tangents at $\mathbf{W}^{(t)}$, we then obtain the following convex optimization problem for each iteration of CCCP

$$\begin{aligned}&\min_{\mathbf{W},\boldsymbol{\eta},\boldsymbol{\xi}}\frac{\lambda}{2}\|\mathbf{W}\|^2+\frac{1}{L}\sum_{j=1}^{L}\eta_j+\frac{\delta}{UC}\sum_{i\in\mathcal{U}}\sum_{z_i=1}^{C}\xi_{iz_i}\\ &s.t.\\ &\forall j,\forall s_{j1},s_{j2}\in\mathcal{Y},s_{j1}\neq s_{j2},l_j=1\ (\mathbf{ML}):\\ &\mathbf{W}^T\Phi\left(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1}^{\mathcal{M}(t)},z_{j2}^{\mathcal{M}(t)}\right)-\mathbf{W}^T\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},s_{j1},s_{j2})\\ &\quad\ge 1-\eta_j,\quad \eta_j\ge 0,\\ &\forall j,\forall s_{j1},s_{j2}\in\mathcal{Y},s_{j1}=s_{j2},l_j=-1\ (\mathbf{CL}):\\ &\mathbf{W}^T\Phi\left(\mathbf{x}_{j1},\mathbf{x}_{j2},z_{j1}^{\mathcal{C}(t)},z_{j2}^{\mathcal{C}(t)}\right)-\mathbf{W}^T\Phi(\mathbf{x}_{j1},\mathbf{x}_{j2},s_{j1},s_{j2})\\ &\quad\ge 1-\eta_j,\quad \eta_j\ge 0,\\ &\forall i\in\mathcal{U},\forall z_i\in\mathcal{Y}:\\ &\mathbf{W}^T\Phi\left(\mathbf{x}_i,y_i^{(t)}\right)-\mathbf{W}^T\Phi(\mathbf{x}_i,z_i)\ge 1-\xi_{iz_i},\xi_{iz_i}\ge 0.\end{aligned} \tag{17}$$

Starting with an initial guess of $\mathbf{W}^{(0)}$, we then solve (17) for $\mathbf{W}^{(t+1)}$, and the iteration continues until convergence.

Good initialization for $\mathbf{W}^{(0)}$ is critical because the original problem in (6) is nonconvex, and the CCCP only finds a local optima. We therefore provide a heuristic method for the initialization. Since MMC computes C scores for each sample using $\mathbf{W}$, such a C-dimensional score vector might be viewed as an embedding coordinate in the C dimensional space. In this space, it is expected that the *cannot-link* pairs of samples should be as far as possible, while the *must-link* pairs should be as close as possible. To this end, we try to find a projection matrix $\mathbf{A}\in\mathbb{R}^{d\times C}$ to map the data into such space by solving

$$\max_{\mathbf{A}} \frac{1}{L_c} \sum_{j:l_j=-1} \left\|\mathbf{A}^T\mathbf{x}_{j1} - \mathbf{A}^T\mathbf{x}_{j2}\right\|^2 = Trace\left(\mathbf{A}^T\mathbf{S}_c\mathbf{A}\right),$$
$$s.t. \quad (18)$$
$$\frac{1}{L_m} \sum_{j:l_j=1} \left\|\mathbf{A}^T\mathbf{x}_{j1} - \mathbf{A}^T\mathbf{x}_{j2}\right\|^2 = Trace\left(\mathbf{A}^T\mathbf{S}_m\mathbf{A}\right) = 1,$$

provided that the samples have been centered, where

$$\mathbf{S}_c = \frac{1}{L_c} \sum_{j:l_j=-1} (\mathbf{x}_{j1} - \mathbf{x}_{j2})(\mathbf{x}_{j1} - \mathbf{x}_{j2})^T, \quad (19)$$

$$\mathbf{S}_m = \frac{1}{L_m} \sum_{j:l_j=1} (\mathbf{x}_{j1} - \mathbf{x}_{j2})(\mathbf{x}_{j1} - \mathbf{x}_{j2})^T. \quad (20)$$

By Lagrange method, $\mathbf{A}$ can be easily solved, i.e., the columns of optimal $\mathbf{A}$ should be the C eigenvectors corresponding to the C largest eigenvalues of $(\mathbf{S}_m)^{-1}\mathbf{S}_c$. Ultimately, the columns of $\mathbf{A}$ are concatenated as the initial guess of $\mathbf{W}^{(0)}$.

The CCCP algorithm for solving (6) is summarized in Algorithm 1. We stop the algorithm when the *relative* difference between objective values of the quadratic program (17) is less than ϵ_1 in two successive iterations, which means the current objective value is only larger than $(1-\epsilon_1)$ of the objective value in the last iteration. In this paper, ϵ_1 is set at 0.01. In order to further reduce the local minima, we set δ at 0 in the first three iterations, i.e., the margin violation for the unconstrained samples is ignored in the first three CCCP iterations. In other words, we try to find a good initial model with the more reliable supervisory information (i.e., the pairwise constraints) alone before using the unconstrained data. In fact, such an "annealing-like" heuristic has also been taken in [28], [29]. Next, we will describe how to optimize the problem (17) by the subgradient projection procedure.

Algorithm 1. The complete algorithm for pairwise constrained MMC using the CCCP

input: $\{\mathbf{x}_i\}_{i\in\mathcal{U}}, \{(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j)\}_{j=1}^{L}, \lambda, \delta_0$, number of clusters C
Solve (18) for $\mathbf{A}$, concatenate its columns to form $\mathbf{W}^{(0)}$;
repeat
  if $t = 0, 1, 2$ **then**
    $\delta = 0$;
  else
    $\delta = \delta_0$;
  end
  Find $y_i^{(t)}$ by (11), $\forall i \in \mathcal{U}$;
  Find $(z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)})$ by (15), $\forall j, l_j = 1$ (**ML**);
  Find $(z_{j1}^{\mathcal{C}(t)}, z_{j2}^{\mathcal{C}(t)})$ by (16), $\forall j, l_j = -1$ (**CL**);
  Solve (17) for $\mathbf{W}^{(t+1)}$ by Algorithm 2;
until *stopping criterion is satisfied (suppose it stops in the $\overline{T}$-th iteration)*;
output: $\widehat{\mathbf{W}} = \mathbf{W}^{(\overline{T})}$ and the cluster index for each point $\hat{y}_i^* = \arg\max_{y\in\mathcal{Y}} \widehat{\mathbf{W}}^T\Phi(\mathbf{x}_i, y), i = 1, \ldots, n$.

3.2 The Subgradient Projection Method for Quadratic Program

The constrained optimization problem in (17) is a standard convex quadratic program. However, the first inequity and the second inequity in (17) impose all the possible combinations of two clusters for each pairwise constraint. Furthermore, the third inequity in (17) will introduce a constraint for every possible candidate cluster label on each unconstrained sample. The number of constraints therefore scales exponentially. In general, the off-the-shelf packages for convex programming are unable to tackle this optimization. Under such circumstances, we convert the constrained optimization in (17) into an equivalent unconstrained one because it is generally easier to implement the unconstrained optimization than the constrained one. First, the slack variables in (17) are resolved analogously to those in SVM [30], for example,

$$\begin{aligned}\eta_{j:l_j=1} &= \max\Bigg\{\max_{s_{j1}\neq s_{j2}} \Big\{1 - \Big[\mathbf{W}^T\Phi\Big(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)}\Big) \\ &\quad - \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\Big]\Big\}, 0\Bigg\} \\ &= \max_{s_{j1}\neq s_{j2}} \Big\{1 - \Big[\mathbf{W}^T\Phi\Big(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)}\Big) \\ &\quad - \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\Big]\Big\}_+ \\ &= \Bigg\{1 - \Bigg[\mathbf{W}^T\Phi\Big(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)}\Big) \\ &\quad - \max_{s_{j1}\neq s_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\Bigg]\Bigg\}_+.\end{aligned}$$

Similarly, we can obtain

$$\begin{aligned}\eta_{j:l_j=-1} &= \Bigg\{1 - \Bigg[\mathbf{W}^T\Phi\Big(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{C}(t)}, z_{j2}^{\mathcal{C}(t)}\Big) \\ &\quad - \max_{s_{j1}=s_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\Bigg]\Bigg\}_+, \\ \xi_{iz_i} &= \left\{1 - \left[\mathbf{W}^T\Phi\left(\mathbf{x}_i, y_i^{(t)}\right) - \mathbf{W}^T\Phi(\mathbf{x}_i, z_i)\right]\right\}_+.\end{aligned}$$

Then, (17) can be transformed to the following simple convex quadratic optimization problem without constraints

$$\begin{aligned}\min_{\mathbf{W}} f(\mathbf{W}) &= \frac{\lambda}{2}\|\mathbf{W}\|^2 \\ &+ \frac{1}{L}\sum_{j:l_j=1} \Bigg\{1 - \Bigg[\mathbf{W}^T\Phi\Big(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)}\Big) \\ &- \max_{s_{j1}\neq s_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\Bigg]\Bigg\}_+ \\ &+ \frac{1}{L}\sum_{j:l_j=-1} \Bigg\{1 - \Bigg[\mathbf{W}^T\Phi\Big(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{C}(t)}, z_{j2}^{\mathcal{C}(t)}\Big) \\ &- \max_{s_{j1}=s_{j2}} \mathbf{W}^T\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2})\Bigg]\Bigg\}_+ \\ &+ \frac{\delta}{UC}\sum_{i\in\mathcal{U}}\sum_{z_i} \left\{1 - \left[\mathbf{W}^T\Phi\left(\mathbf{x}_i, y_i^{(t)}\right) - \mathbf{W}^T\Phi(\mathbf{x}_i, z_i)\right]\right\}_+.\end{aligned} \quad (21)$$

We employ an efficient subgradient projection optimization method for solving the convex problem (21), which has

been widely adopted for its efficiency and effectiveness, e.g., see [31], [32], [23]. First, we need to present the following theorem:

Theorem 1. *The optimal solution of (17) is in the set*

$$\mathcal{B} = \left\{ \mathbf{W} : \|\mathbf{W}\| \leq \rho = \sqrt{\frac{1+\delta}{\lambda}} \right\}. \tag{22}$$

Proof. See Appendix 1, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.68. □

Starting with a random vector $\mathbf{W}_0$ satisfying $\|\mathbf{W}_0\| \leq \rho$, we then take the following simple update rule for optimizing (21)

$$\mathbf{W}_{r+\frac{1}{2}} = \mathbf{W}_r - \mu_r \nabla_r, \tag{23}$$

$$\mathbf{W}_{r+1} = \Pi\left[\mathbf{W}_{r+\frac{1}{2}}\right], \tag{24}$$

where $\nabla_r = \frac{\partial f(\mathbf{W})}{\partial \mathbf{W}}|_{\mathbf{W}_r}$ is a subgradient of $f(\mathbf{W})$ evaluated at $\mathbf{W}_r$. $\mu_r = 1/\lambda r$ is the step size in the rth iteration. The operator $\Pi[\cdot]$ projects a vector onto the set $\mathcal{B}$, i.e., $\Pi[\mathbf{\Theta}] = \arg\min_{\mathbf{\Omega}\in\mathcal{B}} \|\mathbf{\Omega} - \mathbf{\Theta}\|_2^2$. The convergence of the iterative method of this form to the optimal solution is guaranteed [33] provided that the gradient step size is sufficiently small.

Specifically, according to [34], the subgradient of $f(\mathbf{W})$ is calculated as follows:

$$\begin{aligned}
\nabla_r = \lambda \mathbf{W}_r &+ \frac{1}{L} \sum_{j\in\mathcal{M}^v} \left[\Phi\left(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}^{\mathcal{M}^v}, s_{j2}^{\mathcal{M}^v}\right) \right. \\
&\left. - \Phi\left(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)}\right) \right] \\
&+ \frac{1}{L} \sum_{j\in\mathcal{C}^v} \left[\Phi\left(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}^{\mathcal{C}^v}, s_{j2}^{\mathcal{C}^v}\right) \right. \\
&\left. - \Phi\left(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{C}(t)}, z_{j2}^{\mathcal{C}(t)}\right) \right] \\
&+ \frac{\delta}{UC} \sum_{i, z_i \in \mathcal{UZ}^v} \left[\Phi(\mathbf{x}_i, z_i) - \Phi\left(\mathbf{x}_i, y_i^{(t)}\right) \right],
\end{aligned} \tag{25}$$

where we define

$$\begin{aligned}
\mathcal{M}^v = \Big\{ j : l_j = 1 \Big| & \mathbf{W}_r^T \Phi\left(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)}\right) \\
& - \max_{s_{j1} \neq s_{j2}} \mathbf{W}_r^T \Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2}) < 1 \Big\}, \\
\mathcal{C}^v = \Big\{ j : l_j = -1 \Big| & \mathbf{W}_r^T \Phi\left(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}^{\mathcal{C}(t)}, z_{j2}^{\mathcal{C}(t)}\right) \\
& - \max_{s_{j1} = s_{j2}} \mathbf{W}_r^T \Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2}) < 1 \Big\}, \\
\mathcal{UZ}^v = \Big\{ i \in \mathcal{U}, z_i \in \mathcal{Y} \Big| & \mathbf{W}_r^T \Phi\left(\mathbf{x}_i, y_i^{(t)}\right) - \mathbf{W}_r^T \Phi(\mathbf{x}_i, z_i) < 1 \Big\},
\end{aligned}$$

and

$$\begin{aligned}
\left(s_{j1}^{\mathcal{M}^v}, s_{j2}^{\mathcal{M}^v}\right) &= \arg \max_{s_{j1} \neq s_{j2} : l_j = 1} \mathbf{W}_r^T \Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2}), \\
\left(s_{j1}^{\mathcal{C}^v}, s_{j2}^{\mathcal{C}^v}\right) &= \arg \max_{s_{j1} = s_{j2} : l_j = -1} \mathbf{W}_r^T \Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, s_{j1}, s_{j2}).
\end{aligned}$$

Thereby, in the rth iteration, we first identify the sets $\mathcal{M}^v$ and $\mathcal{C}^v$ for pairwise constraints, and the set $\mathcal{UZ}^v$ for unconstrained samples, on which $\mathbf{W}_r$ violates the hard margin. Then, the subgradient is calculated with these sets. Consequently, $\mathbf{W}_r$ is updated along the gradient descent direction.

To ensure that the weight vector $\mathbf{W}_{r+\frac{1}{2}}$ keeps in the parameter space, in which the optimum resides, we need to project the updated $\mathbf{W}_{r+\frac{1}{2}}$ back to the set $\mathcal{B}$. Since $\mathcal{B}$ is an origin-centered ball of radius ρ, the projection of $\mathbf{W}_{r+\frac{1}{2}}$ onto set $\mathcal{B}$ amounts to scaling $\mathbf{W}_{r+\frac{1}{2}}$ by $\min\{1, \rho/\|\mathbf{W}_{r+\frac{1}{2}}\|\}$. As shown in [32], the projection step can effectively accelerate the convergence in comparison with the standard subgradient descent without the projection step.

The main steps of the iterative subgradient descent and projection procedure are summarized in Algorithm 2. The following convergence property of Algorithm 2 can be obtained by taking a derivation similar to that in [32]. The mathematical proof is therefore omitted here.

Algorithm 2. Solving (17) by subgradient projection

input: $\{(\mathbf{x}_i, y_i^{(t)})\}_{i\in\mathcal{U}}, \{(z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)}, l_j = 1)\}_{j=1}^{L}, \{(z_{j1}^{\mathcal{C}(t)}, z_{j2}^{\mathcal{C}(t)}, l_j = -1)\}_{j=1}^{L}, \{(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j)\}_{j=1}^{L}, \lambda, \delta_0$, number of clusters C

Initialize $\mathbf{W}_0$ ($r = 0$) randomly subject to $\|\mathbf{W}_0\| \leq \rho$;

repeat

Calculate ∇_r by (25);

$\mu_r = \frac{1}{\lambda r}$;

$\mathbf{W}_{r+\frac{1}{2}} = \mathbf{W}_r - \mu_r \nabla_r$;

$\mathbf{W}_{r+1} = \min\{1, \rho/\|\mathbf{W}_{r+\frac{1}{2}}\|\} \times \mathbf{W}_{r+\frac{1}{2}}$;

until *stopping criterion is satisfied (suppose it stops in the R-th iteration)*;

output $\mathbf{W}_R$

Theorem 2. *Algorithm 2 converges to a σ-accurate solution $\widetilde{\mathbf{W}}$ for (17) (or equivalently (21)) after $\mathcal{O}(\bar{R}^2/\lambda\sigma)$ iterations, where $\widetilde{\mathbf{W}}$ satisfies $f(\widetilde{\mathbf{W}}) \leq \min_{\mathbf{W}} f(\mathbf{W}) + \sigma$, and*

$$\begin{aligned}
\bar{R} = 2 \max \Big\{ & \max_{\{(\mathbf{x}_{j1}, \mathbf{x}_{j2}, l_j)\}_{j=1}^{L}, z_{j1}, z_{j2} \in \mathcal{Y}} \|\Phi(\mathbf{x}_{j1}, \mathbf{x}_{j2}, z_{j1}, z_{j2})\|, \\
& \delta \cdot \max_{i\in\mathcal{U}} \|\mathbf{x}_i\| \Big\}.
\end{aligned} \tag{26}$$

In practice, we stop Algorithm 2 as soon as $\|\mathbf{W}_r - \mathbf{W}_{r+1}\| / \max\{\|\mathbf{W}_r\|, \|\mathbf{W}_{r+1}\|\} \leq \epsilon_2$. In our later experiments, ϵ_2 is set at 0.01.

3.3 Time Complexity Analysis

This section analyzes the time complexity of the proposed pairwise constrained MMC algorithm. To solve (17) using Algorithm 2, the dominant computational cost is the calculation of the subgradient whose complexity is $\mathcal{O}(nd)$. Here, d denotes either the dimensionality of the sample in original input space, or that of the coordinate for each sample in the kernel PCA basis according to a kernel function. Furthermore, from Theorem 2, it can be seen that the number of iterations of Algorithm 2 is independent of n and d. Hence, Algorithm 2 takes the time $\mathcal{O}(nd)$ to converge.

In the complete algorithm, i.e., Algorithm 1, the initialization by finding the projection matrix takes $\mathcal{O}(d^3)$ time. In each iteration of Algorithm 1, it only involves the inner product operation when finding $y_i^{(t)}$, $(z_{j1}^{\mathcal{M}(t)}, z_{j2}^{\mathcal{M}(t)})$ and $(z_{j1}^{\mathcal{C}(t)}, z_{j2}^{\mathcal{C}(t)})$, which scales with time $\mathcal{O}(nd)$. Hence, the overall complexity of the proposed pairwise constrained MMC in Algorithm 1 is $\mathcal{O}(d^3 + \overline{T}nd)$, where $\overline{T}$ is the number of iterations for the CCCP to converge. In our experiments, it was found that $\overline{T}$ never exceeded 50.

4 Evaluation

In this section, we evaluated the accuracy and efficiency of our proposed algorithm on a couple of real-world data sets. Moreover, the scalability of the proposed method was investigated as well. In addition, we examined the generalization capability of the proposed approach, i.e., the accuracy performance on the out-of-sample data points. All the experiments were conducted with Matlab 7.0 on a 2.4 GHz Intel Core 2 PC running Windows XP with 3.25 G main memory.

4.1 Evaluation Data Sets

The experiments were performed on the data sets from the UCI repository[1] (`pendigit389`, `letterIJL`, `vehicle`, `ionosphere`, `sonar`, `optdigit odd versus even`, `statlogSegment`, `mfea-fac`, `magic`), the MNIST handwritten digit database[2] (`mnist0689`), the USPS handwritten digit database[3] (`uspst`), the COIL image database[4] (`COIL3`), the brain-computer interface database[5] (`BCI`), and the document database of the CLUTO toolkit[6] (`sports`, `ohscal`). These data sets provide a good representation of different characteristics: number of samples ranges from 216 to 19,020, dimensionalities from 10 to 1,024, and number of classes from 2 to 10. A summary of all the data sets used in this paper is shown in Table 1.

The data sets `pendigit389` and `letterIJL` have been used in [6]. Three confusing classes ("3, 8, 9") and ("I, J, L") are chosen from the pendigit and letter databases, respectively. Moreover, both `pendigit389` and `letterIJL` consist of 10 percent randomly sampled data points from the whole data sets in UCI repository. The MNIST handwritten digits database is relatively large for most algorithms, we followed [18] and randomly selected 200 samples from each of the corresponding classes ("0, 6, 8, 9") to form the `mnist0689` data set. Fig. 3 shows some samples from this data set. It can be seen that the samples from different classes appear to be similar. Further, some samples are overlapped and prone to be incorrectly classified, e.g., the ninth digit "6" in the second row and the third digit "8" in the third row. In the `COIL3` data set, the data are gray images for three classes of cars which look much alike (see Fig. 3), indicating a severe overlap among the samples from the three classes. The samples were scaled to $[0, 1]$ for these two image data sets. The semi-supervised clustering on the optdigit `odd vs.even` data set is an artificial task with two classes (handwritten digits "1, 3, 5, 7, 9" versus "0, 2, 4, 6, 8"), it simulates the scenario, where the user provides the pairwise constraints for producing desired clusters. Both the `mfea-fac` and `uspst` data sets contain the handwritten digits "0-9" but with the different features, i.e., profile correlations and pixels, respectively. These digit samples share a large number of features, thus usually leading to a difficult partitioning task. The `sports` data set contains the articles about baseball, hockey, basketball, bicycling, boxing, football, and golfing from the San Jose Mercury newspapers. The `ohscal` data set was from the OHSUMED collection [35]. It contains 11,162 documents from the following ten categories: antibodies, carcinoma, DNA, in vitro, molecular sequence data, pregnancy, prognosis, receptors, risk factors, and tomography. It is generally known that the articles from different subtopics have a lot of words in common. Furthermore, for these two text data sets both in bag-of-words representation, we selected the top 1,000 words by information gain with class labels for the following experiments. In a word, there is generally severe overlapping among the samples from different classes in most of the data sets used in this paper, which are thus quite appropriate for evaluating the robustness of the semi-supervised clustering algorithms.

TABLE 1
The Data Sets Used in Experiments

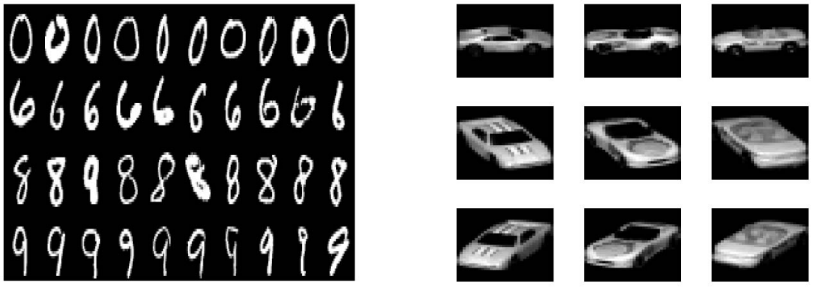
Dataset	#Dimension (d)	#Sample (n)	#Class (C)
small size			
`pendigit389`	16	318	3
`letterIJL`	16	227	3
`vehicle`	18	846	4
`ionosphere`	34	351	2
`sonar`	60	208	2
`BCI`	117	400	2
`mnist0689`	784	800	4
`COIL3`	1024	216	3
medium size			
`odd vs.even`	64	1797	2
`statlogSegment`	19	2310	7
`mfea-fac`	216	2000	10
`uspst`	256	2007	10
large size			
`magic`	10	19020	2
`sports`	1000	8580	7
`ohscal`	1000	11162	10

4.2 Evaluation Metrics

To evaluate the effectiveness of clustering algorithms, we used the clustering accuracy (ACC) and the normalized mutual information (NMI) in this paper. Following the strategy in [16], [18], we first take a set of labeled data, remove the labels and run the clustering algorithms. Then, we label each of the resulting clusters with the majority class according to the original training labels. Finally, the clustering accuracy is defined as the matching degree between the obtained labels and the original true labels

$$ACC = \frac{\sum_{i=1}^{n} I(\hat{t}_i = t_i)}{n}, \tag{27}$$

1. http://www.ics.uci.edu/~mlearn/MLRepository.html.
2. http://www.cs.toronto.edu/~roweis/data.html.
3. http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/data.html.
4. http://www.kyb.tuebingen.mpg.de/bs/people/chapelle/lds/.
5. http://www.kyb.tubingen.mpg.de/ssl-book.
6. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download.



(a) (b)

Fig. 3. Sample images from (a) `mnist0689` and (b) `COIL3` data sets.

where $\hat{t}_i$ is the label obtained by the above steps for $\mathbf{x}_i$, and t_i is the ground-truth label. The NMI is defined by Strehl and Ghosh [36]

$$NMI = \frac{\sum_{i=1}^{C}\sum_{j\in\mathcal{C}'} n_{i,j}\log(\frac{n\cdot n_{i,j}}{n_i\tilde{n}_j})}{\sqrt{(\sum_{i=1}^{C} n_i\log\frac{n_i}{n})(\sum_{j\in\mathcal{C}'}\tilde{n}_j\log\frac{\tilde{n}_j}{n})}}, \qquad (28)$$

where n is the number of samples in the data set, n_i denotes the number of data contained in the cluster $\mathcal{C}_i(1 \le i \le C)$, $\tilde{n}_j$ is the number of data belong to the jth class ($j \in \mathcal{C}', \mathcal{C}'$ is the set of ground truth classes), and $n_{i,j}$ denotes the number of data that are in the intersection between the cluster $\mathcal{C}_i$ and the jth class. The NMI ranges from 0 to 1. The larger the value it is, the more similar the groupings by clustering and those by the true class labels.

In the experiments, the execution time of the proposed algorithm was also compared with that of the other competing algorithms.

4.3 Competing Algorithms

Besides the proposed algorithm, we also implemented some competing counterparts as well as the baseline methods listed below for comparison.

1. **Kmeans**: This is the traditional K-means algorithm without any pairwise constraints incorporated.
2. **CPMMC**: This is the latest unsupervised maximum margin clustering algorithm proposed in [16].
3. **DCA+Kmeans**: The Discriminative Component Analysis (DCA) [5] first learns a distance metric based on the pairwise constraints. Then, K-means is performed with this metric. It has been shown in [5] that DCA can gain better performance than its counterparts: Relevant Component Analysis (RCA) [4], and Xing's method [3].
4. **MPCKmeans**: This is the well-known semi-supervised clustering algorithm which alternates between the metric learning step and the clustering step [6].
5. **CPCMMC**: This is the only existing work on semi-supervised maximum margin clustering in the literature [18].

The Kmeans and CPMMC served as the baselines. For our method, we found that the performance was relatively insensitive to the value of δ_0, thus it was set at 1 throughout the experiments, and λ was selected in a set of candidates $\{0.01, 0.1, 1, 10, 100\}$, from which the best result was reported. For the other algorithms with free parameters, we also reported their best performance with their parameters chosen from a set of candidates. Moreover, Kmeans, DCA+Kmeans, and MPCKmeans worked on the raw input for all the data sets. For CPMMC, CPCMMC, and the proposed method, all the data sets were preprocessed so that all the features have zero mean. For the `vehicle`, `statlogSegment`, `mfea-fac`, `magic` data sets whose features are of very different scales, we further let all the features have unit standard deviation. For the `COIL3` data set, its sample size is much smaller than its dimensionality. To ease the computation, CPMMC, CPCMMC, and our method worked on the data representation obtained by Kernel PCA, where the linear kernel was used for simplicity, and the reduced dimensionality was set at its sample size. In the experiments, we let the number of clusters be the true number of classes for all the algorithms, although the selection of the number of clusters is a crucial issue, which is, however, beyond the scope of this paper.

For each data set, we evaluated the performance with different numbers of pairwise constraints. Each constraint was generated by randomly selecting a pair of samples. If the samples belong to the same class, a *must-link* constraint was formed. Otherwise, a *cannot-link* constraint was formed. For a fixed number of pairwise constraints, the results were averaged over 20 realizations of different pairwise constraints, which are shown in Figs. 4 and 5. Since the learning of MPCKmeans cannot finish in reasonable time or have memory overflow problem on the `mnist0689`, `COIL3`, `magic`, `sports`, and `ohscal` data sets which have either high dimensionalities or large sample size, we do not include it for comparison on these data sets.

4.4 Evaluation Results

4.4.1 Comparison of Effectiveness

Figs. 4 and 5 show that the proposed constrained MMC can dramatically improve the performance of the baseline Kmeans and CPMMC. Furthermore, it can be seen that the proposed algorithm performs the best in most cases. Actually, the pairwise constraints used are quite sparse. For example, only 10 percent of the data from each class of the small-size `COIL3` data set can generate 211 constraints. This number is more than 100 that is the largest number of pairwise constraints provided for such a data set in the experiment. Also, only 1 percent of the data from each class

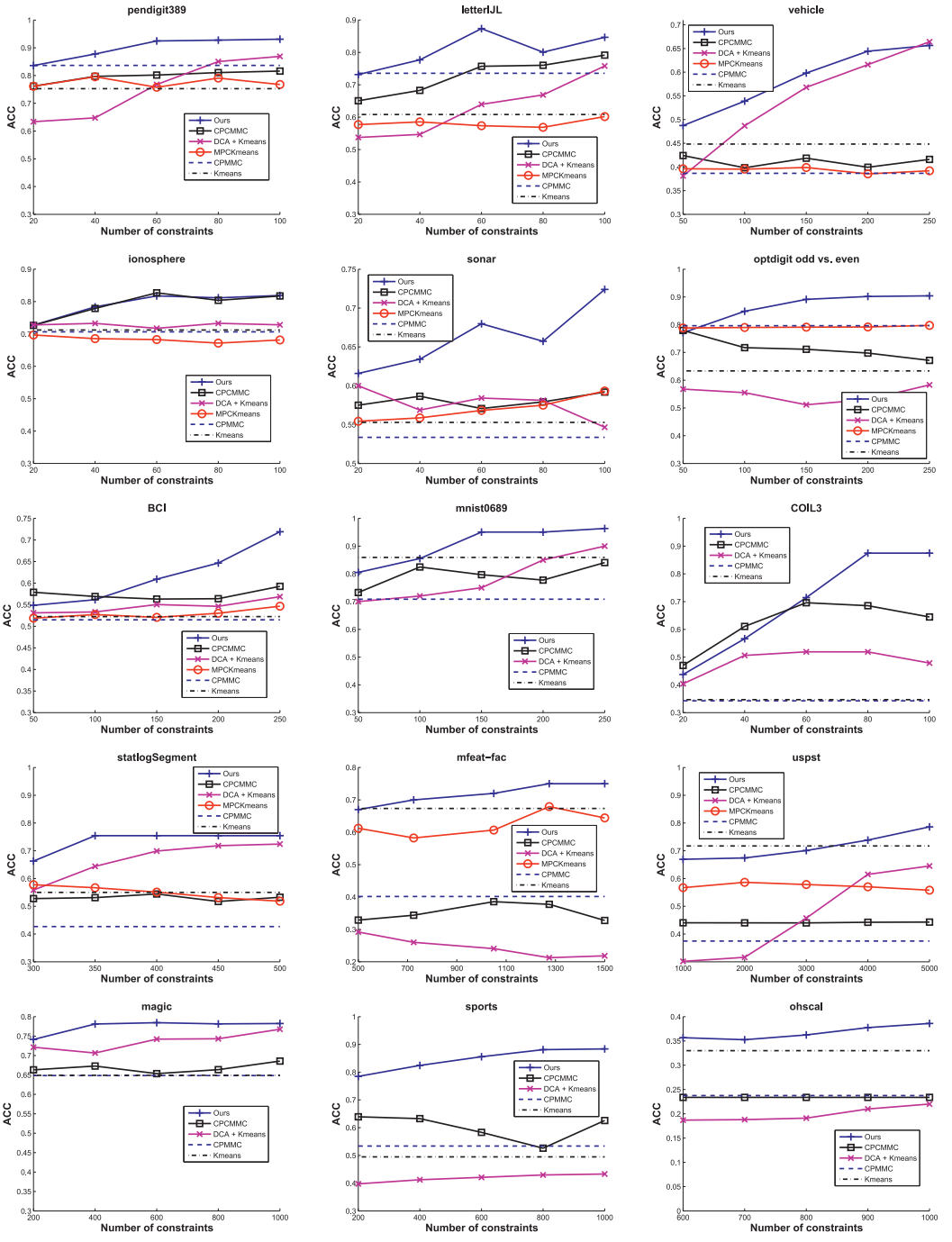


Fig. 4. Comparison of clustering accuracy over the different number of pairwise constraints.

of the large-size `magic` data set can produce 17,766 constraints. Again, this figure is much more than 1,000 that is the largest number of pairwise constraints used on such a data set. It implies that the proposed method indeed propagates the pairwise constraints effectively. The plausible reasons are two-fold. One is that either the violation of margin requirements for the unconstrained samples or the pairwise constraints will incur heavy penalties in our approach, which enables to effectively find the most desired cluster for the *must-link* constraint and two different clusters for the *cannot-link* constraint, while ensuring that the partitioning boundary of each cluster is as far as possible from those of others. The other reason is that the proposed method utilizes the pairwise constraints for initialization, i.e., solving the projection matrix to initialize the $\mathbf{W}^{(0)}$ and carrying out the "annealing-like" heuristic to seek a good initial model. By contrast, the ACC/NMI curves for the other semi-supervised clustering algorithms often show a slow rise on some data sets with the increase of amounts of pairwise constraints, or even fall behind those for the baselines. Specifically, for the metric-based algorithms, i.e., MPCKmeans and DCA+Kmeans, the estimation of metric parameters with few constraints is generally unreliable [6], especially on the high-dimensional data sets (e.g., `mnist0689`, `COIL3`, `mfea-fac`, `uspst`, `sports`, `ohscal`). To obtain an accurate estimation, a large number of pairwise

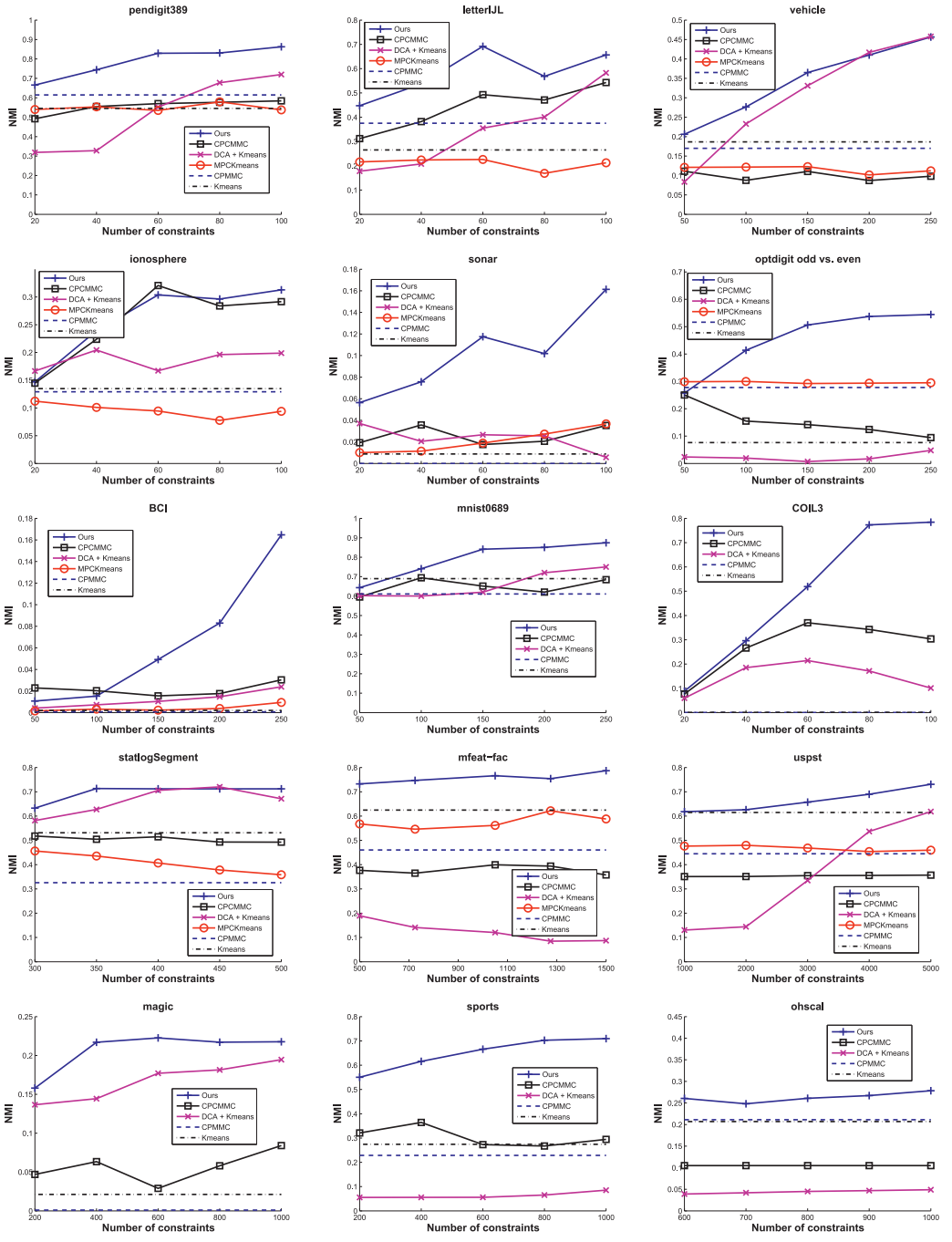


Fig. 5. Comparison of normalized mutual information over the different number of pairwise constraints.

constraints are therefore needed. Although the CPCMMC can improve the performance on some data sets, the improvement is often inferior to that by the proposed method. The reason is that there is severe overlapping in most used data sets (c.f. Section 4.1), but the loss function for violating the *cannot-link* constraints in CPCMMC [18] is less able to effectively discourage the violation of constraints in such scenarios.

4.4.2 Comparison of Efficiency

Fig. 6 presents the average and standard deviation of the CPU time consumption for 20 trials of each semi-supervised clustering algorithm, with different number of pairwise constraints. It can be seen that the proposed method is generally time efficient (about 1 minute on the `magic` data set with 19,020 samples owning 200-1,000 pairwise constraints, and about 20 minutes on the `ohscal` data set with 11,162 samples possessing 600-1,000 constraints). Although it is not always the fastest algorithm on all the data sets, it is much more advantageous than its counterparts in terms of the accuracy. It is also observed that the proposed method worked stably on all the data sets we have tried so far, as indicated by the small deviations of the execution time. By contrast, the other counterparts often showed the large variations of the CPU-time, especially on the large-size data sets in Table 1, e.g., `magic`, `sports`, and `ohscal`.

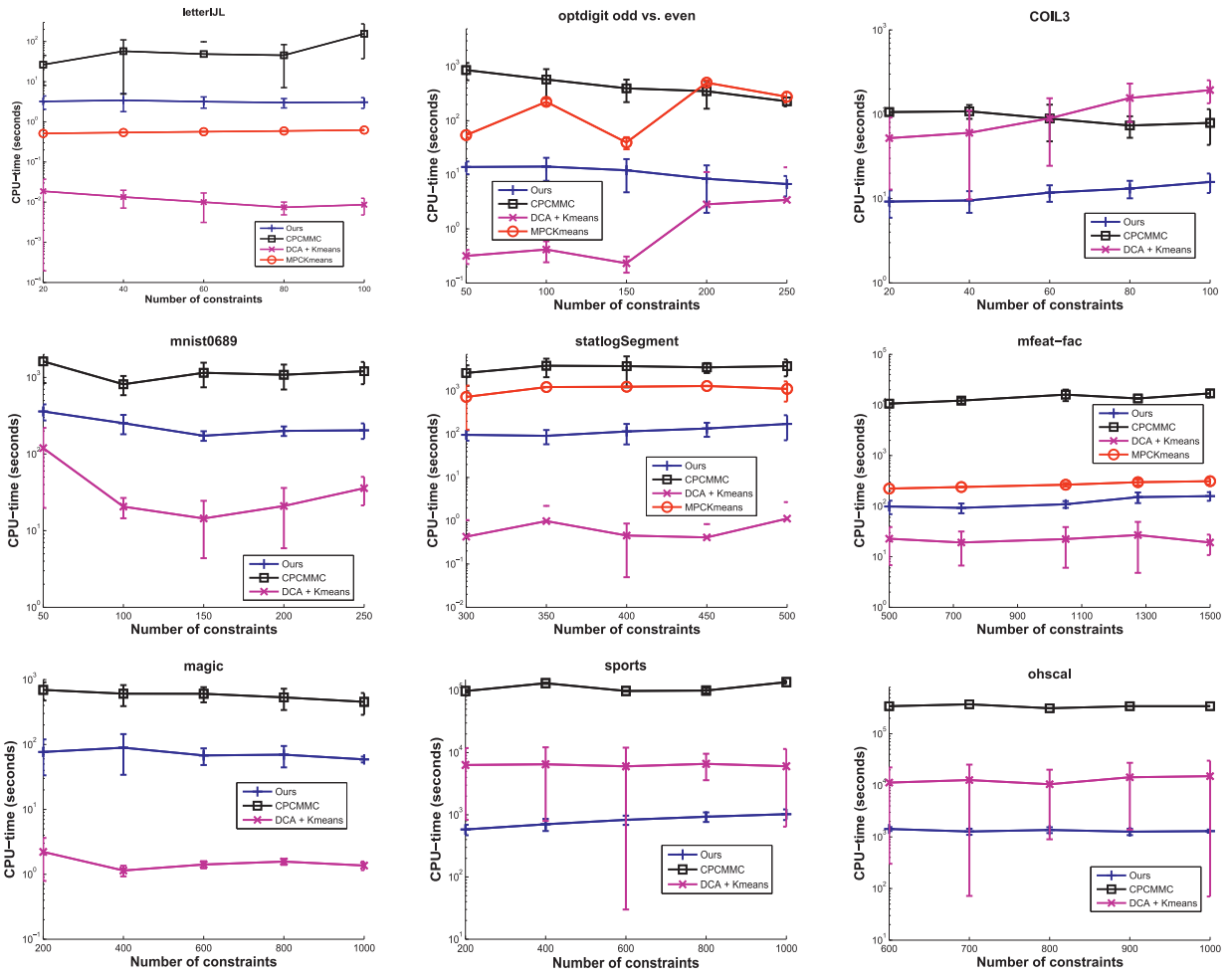


Fig. 6. Comparison of execution time for the semi-supervised clustering algorithms.

4.4.3 Scalability of the Proposed Method

To demonstrate the scalability of the proposed algorithm, we showed its average execution time (the initialization time was ignored) as a function of the sample size on the three large-size data sets over 10 trials in Fig. 7. The value in the parentheses denotes the fixed number of pairwise constraints provided for each data set. The dashed line represents the linear growth $O(n)$. It can been seen that the execution time of the proposed method on each data set roughly shows a linear trend, which is consistent with the time-complexity analysis in Section 3.3. That is, without considering the initialization time, the time complexity of the proposed method scales linearly with the number of samples.

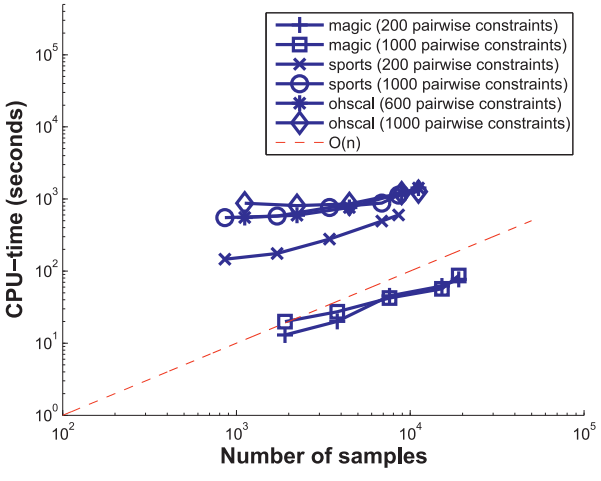


Fig. 7. Execution time of the proposed method over the sample size n.

4.4.4 Generalization Performance

To examine the generalization ability of the proposed algorithm on out-of-sample examples, we used the learned models from the previous experiments to partition the unseen testing samples into the clusters. Fig. 8 reports the average clustering accuracy for out-of-sample examples on `pendigit389`, `letterIJL`, `optdigit odd vs.even`, and `mnist0689` data sets, in which only a small subset was used for training as described in the previous experiments, and the sizes of the unseen testing subsets are 2847, 2036, 3823, and 800, respectively. From Fig. 8, it can be seen that the proposed method again outperforms the counterpart constrained MMC [18] on the out-of-sample examples for these data sets. Moreover, for both our constrained MMC and the one in [18], the clustering accuracy on out-of-sample examples is comparable with that on the training subset. Thus, given a large data set, we may first perform the proposed constrained MMC on a

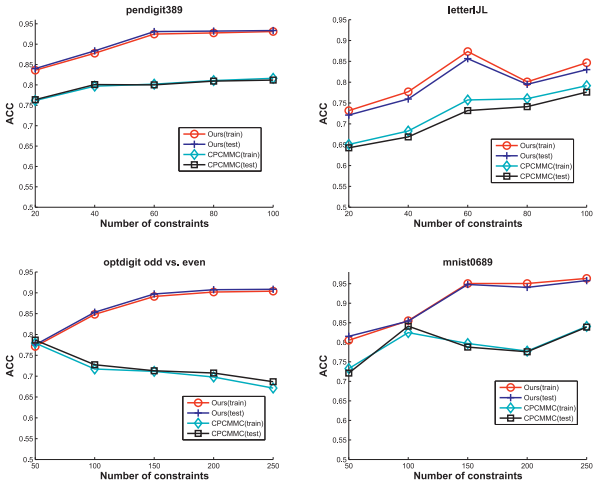


Fig. 8. Comparison of clustering accuracy on out of sample examples.

small subset, and then use the learned model to cluster the remaining points [18], from a practical viewpoint.

5 Conclusion

In this paper, we have proposed a pairwise constrained maximum margin clustering algorithm. A set of loss functions for violating the pairwise constraints is introduced in a soft-margin formulation. In contrast to those used in an existing semi-supervised maximum margin clustering algorithm [18], the proposed loss functions can provide more robust penalization to the violation for the pairwise constraints. To optimize the resulting nonconvex clustering problem, the CCCP has been utilized to decompose it into a sequence of convex quadratic program problems. Each of the convex problems in the CCCP sequence is then solved by an efficient subgradient projection procedure. Experimental results have shown that the proposed constrained MMC algorithm effectively improves the baseline MMC, and is competitive or even better than the preliminary constrained MMC algorithm [18], as well as some typical semi-supervised clustering counterparts, both in accuracy and efficiency. Moreover, we have shown that the execution time of the proposed method scales linearly with the sample size n after its initialization. In the experiments, the pairwise constraints are provided beforehand. In the future work, we plan to actively identify the most informative pairwise constraints for the maximum margin clustering during the clustering process.

Acknowledgments

The work described in this paper was supported by the Research Grant Council of Hong Kong SAR under Project HKBU 210306 & HKBU 210309, the Faculty Research Grant of Hong Kong Baptist University with the Project Code: FRG2/08-09/122, the National Natural Science Foundation of China (61105048, 61104206), the Doctoral Fund of Ministry of Education of China (20100092120012, 20110092120034) and by the Open Fund of Jiangsu Province Key Laboratory for Remote Measuring and Control (YCCK201005). Yiu-Ming Cheung is the corresponding author for this paper.

References

[1] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-Supervised Graph Clustering: A Kernel Approach," *Proc. Int'l Conf. Machine Learning*, pp. 457-464, 2005.

[2] M. Ester, R. Ge, B.J. Gao, Z. Hu, and B. Ben-Moshe, "Joint Cluster Analysis of Attribute Data and Relationship Data: The Connected K-Center Problem," *Proc. SIAM Int'l Conf. Data Mining*, pp. 25-46, 2006.

[3] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell, "Distance Metric Learning with Application to Clustering with Side-Information," *Advances in Neural Information Processing Systems*, vol. 15, pp. 521-528, 2003.

[4] N. Shental, T. Hertz, D. Weinshall, and M. Pavel, "Adjustment Learning and Relevant Component Analysis," *Proc. European Conf. Computer Vision*, pp. 776-792, 2002.

[5] S.C.H. Hoi, W. Liu, M.R. Lyu, and W.Y. Ma, "Learning Distance Metrics with Contextual Constraints for Image Retrieval," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 2072-2078, 2006.

[6] M. Bilenko, S. Basu, and R.J. Mooney, "Integrating Constraints and Metric Learning in Semi-Supervised Clustering," *Proc. Int'l Conf. Machine Learning*, pp. 81-88, 2004.

[7] K. Wagstaff, C. Cardie, and S. Schroedl, "Constrained K-Means Clustering with Background Knowledge," *Proc. Int'l Conf. Machine Learning*, pp. 577-584, 2001.

[8] S. Basu, M. Bilenko, and R.J. Mooney, "A Probabilistic Framework for Semi-Supervised Clustering," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 59-68, 2004.

[9] N. Shental, A. Bar-Hillel, D. Weinshall, "Computing Gaussian Mixture Models with EM Using Equivalence Constraints," *Advances in Neural Information Processing Systems*, vol. 16, pp. 465-472, 2004.

[10] M. Law, A. Topchy, and A.K. Jain, "Model-Based Clustering with Probabilistic Constraints," *Proc. SIAM Int'l Conf. Data Mining*, pp. 641-645, 2005.

[11] Z. Lu and M.A. Carreira-Perpinan, "Constrained Spectral Clustering through Affinity Propagation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[12] N. Kumar and K. Kummamuru, "Semisupervised Clustering with Metric Learning Using Relative Comparisons," *IEEE Trans. Knowledge and Data Eng.*, vol. 20, no. 4, pp. 496-503, Apr. 2008.

[13] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum Margin Clustering," *Advances in Neural Information Processing Systems*, vol. 17, pp. 1537-1544, 2005.

[14] H. Valizadegan and R. Jin, "Generalized Maximum Margin Clustering and Unsupervised Kernel Learning," *Advances in Neural Information Processing Systems*, vol. 19, pp. 1417-1424, 2007.

[15] K. Zhang, I.W. Tsang, and J.T. Kwok, "Maximum Margin Clustering Made Practical," *Proc. Int'l Conf. Machine Learning*, pp. 1119-1126, 2007.

[16] B. Zhao, F. Wang, and C. Zhang, "Efficient Multiclass Maximum Margin Clustering," *Proc. Int'l Conf. Machine Learning*, pp. 1248-1255, 2008.

[17] Y.F. Li, I.W. Tsang, J.T. Kwok, and Z.H. Zhou, "Tighter and Convex Maximum Margin Clustering," *Proc. Int'l Conf. Artificial Intelligence and Statistics*, pp. 344-351, 2009.

[18] Y. Hu, J. Wang, N. Yu, and X.S. Hua, "Maximum Margin Clustering with Pairwise Constraints," *Proc. IEEE Int'l Conf. Data Mining*, pp. 253-262, 2008.

[19] S.C.H. Hoi, R. Jin, and M.R. Lyu, "Learning Nonparametric Kernel Matrices from Pairwise Constraints," *Proc. Int'l Conf. Machine Learning*, pp. 361-368, 2007.

[20] K. Crammer and Y. Singer, "On the Algorithmic Implementation of Multiclass Kernel-Based Vector Machines," *J. Machine Learning Research*, vol. 2, pp. 265-292, 2002.

[21] L. Xu and D. Schuurmans, "Unsupervised and Semi-Supervised Multi-Class Support Vector Machines," *Proc. Nat'l Conf. Artificial Intelligence*, pp. 904-910, 2005.

[22] R. Yan, J. Zhang, J. Yang, and A. Hauptmann, "A Discriminative Learning Framework with Pairwise Constraints for Video Object Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 578-593, Apr. 2006.

[23] N. Nguyen and R. Caruana, "Improving Classification with Pairwise Constraints: A Margin-Based Approach," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases*, pp. 113-124, 2008.

[24] T. Joachims, "Training Linear SVMs in Linear Time," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 217-226, 2006.
[25] A.L. Yuille and A. Rangarajan, "The Concave-Convex Procedure," *Neural Computation,* vol. 15, no. 4, pp. 915-936, 2003.
[26] A.J. Smola, S.V.N. Vishwanathan, and T. Hofmann, "Kernel Methods for Missing Variables," *Proc. Int'l Workshop Artificial Intelligence and Statistics,* pp. 325-332, 2005.
[27] K. Zhang, I.W. Tsang, and J.T. Kwok, "Maximum Margin Clustering Made Practical," *IEEE Trans. Neural Networks,* vol. 20, no. 4, pp. 583-596, Apr. 2009.
[28] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large Scale Transductive SVMs," *J. Machine Learning Research,* vol. 7, pp. 1687-1712, 2006.
[29] M. Karlen, J. Weston, A. Erkan, and R. Collobert, "Large Scale Manifold Transduction," *Proc. Int'l Conf. Machine Learning,* pp. 448-455, 2008.
[30] A. Zien, U. Brefeld, and T. Scheffer, "Transductive Support Vector Machines for Structured Variables," *Proc. Int'l Conf. Machine Learning,* pp. 1183-1190, 2007.
[31] K. Weinberger, J. Blitzer, and L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Advances in Neural Information Processing Systems,* vol. 18, pp. 1473-1480, 2006.
[32] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal Wstimated Sub-Gradient Solver for SVM," *Proc. Int'l Conf. Machine Learning,* pp. 807-814, 2007.
[33] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge Univ. Press, 2004.
[34] D.P. Bertsekas, *Convex Analysis and Optimization.* Athena Scientific, 2003.
[35] W. Hersh, C. Buckley, T.J. Leone, and D. Hickam, "OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research," *Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 192-201, 1994.
[36] A. Strehl and J. Ghosh, "Cluster Ensembles-A Knowledge Reuse Framework for Combining Multiple Partitions," *J. Machine Learning Research,* vol. 3, pp. 583-617, 2003.

Hong Zeng received the PhD degree in computer science from Hong Kong Baptist University, in 2010. He is currently with the Robotic Sensor and Control Laboratory (RSCL) in the School of Instrument Science and Engineering, Southeast University, China. His research interests are in the areas of pattern recognition, machine learning and data mining. He is a member of the IEEE.

Yiu-Ming Cheung (SM'06) received the PhD degree from the Department of Computer Science and Engineering at the Chinese University of Hong Kong in 2000. Currently, he is an associate professor in the Department of Computer Science at Hong Kong Baptist University. His research interests include machine learning, information security, signal processing, pattern recognition, and data mining. He is the founding chair of Computational Intelligence Chapter of IEEE Hong Kong Section. Also, he is a senior member of the IEEE and the ACM. More details can be found at: http://www.comp.hkbu.edu.hk/~ymc.