International Journal of Computational Intelligence Systems, Vol.3, No. 6 (December, 2010), 770-785

# Clustering with Instance and Attribute Level Side Information

**Jinlong Wang [1,3] , Shunyao Wu [1] , Gang Li [2]**

[1] *School of Computer Engineering, Qingdao Technological University, Qingdao, 266033, China*

*E-mail: wangjinlong@gmail.com; shunyaowu@gmail.com*

[2] *School of Information Technology, Deakin University, Victoria, Australia*

*E-mail: gang.li@deakin.edu.au*

[3] *Medical College of Qingdao University, Qingdao, 266021, China*

Received: 16-02-2010
Accepted: 26-10-2010

**Abstract**

Selecting a suitable proximity measure is one of the fundamental tasks in clustering. How to effectively utilize all available side information, including the instance level information in the form of pair-wise constraints, and the attribute level information in the form of attribute order preferences, is an essential problem in metric learning. In this paper, we propose a learning framework in which both the pair-wise constraints and the attribute order preferences can be incorporated simultaneously. The theory behind it and the related parameter adjusting technique have been described in details. Experimental results on benchmark data sets demonstrate the effectiveness of proposed method.

*Keywords:* Data mining, Clustering, Semi-supervised learning, Constraints.

## 1. Introduction

Clustering, partitioning data into sensible groupings according to measured or perceived intrinsic characteristics or similarity, is one of the most fundamental unsupervised data mining tasks [1,2,3,4]. In the past decades, clustering methods have been successfully applied in a variety of applications across a wide range of fields, including computer vision, system biology, and e-business, *etc*. In general, clustering is a subjective process which focuses on finding optimal clusterings according to some specific distance of similarity measures. However, specifying appropriate similarity measures is usually difficult because of its dependency on human expertise.

Recently, the topic of semi-supervised clustering [5,6,7,8,9,10] has attracted a lot of research effort, and it aims to utilize some kinds of side-information to improve the accuracy in clustering. One popular type of side-information is in the form of *Pairwise Link Constraints*, which can be further divided into the *must-link* constraints (instances $i$ and $j$ are in the same cluster) and the *cannot-link* constraints (instances $i$ and $j$ belong to different clusters). Based on such information, the clustering objective function can be modified so that it includes satisfaction of constraints, enforcing constraints during the clustering process [11,12,13,14,15,16]. A second line of re-

search, to which this work belongs, focuses on learning a suitable metric from the dataset augmented by some side-information, relevant to the task at hand. Some recent research sought to address this problem is usually referred to as *Metric Learning* [17,18,19,20]. Xing et al. [17] proposed to learn a *Mahalanobis* metric using the *Pair-wise Link Constraints*, before performing clustering with constraints. Their proposed method is based on posting metric learning as a combination of gradient descent and iterative projection to solve a convex nonlinear optimization problem. Instead of using an iterative algorithm as in the method [17], Bar-Hillel et al. [18] proposed a more efficient, non-iterative algorithm called the Relevant Component Analysis (RCA) algorithm to learn a Mahalanobis metric. More recently, Halkidi et al. [19] proposed a framework for the learning of a weighted Euclidean distance, based on the *Pair-wise Link Constraints* and the cluster validity criteria. In addition, many researches utilized instance-level information in the form of pairwise constraints to assist document clustering [5,21,22,23,24].

All the aforementioned approaches aim to improve the clustering accuracy by utilizing the instance-level side-information, while the attribute-level side-information has largely been overlooked. A different scenario, in which attribute-level side-information is a natural source of training data, occurs when we wish to identify a different clustering criterion from the original one but equally good in terms of the objective clustering evaluation. For example, consider clustering loan applications to determine a method to identify risky loans but the clusters fall along racial lines, the banks may wish to find an alternative criterion but with equally good clustering results. In this situation, existing clustering results can be used as a good resource for instance-level side-information, while the experts' domain preferences of some attributes over the others will be used as the attribute-level side information. In document clustering, attribute-level information in the form of keywords can be obtained from important parts, such as *Title* and *Keywords* [25], or by some methods from keywords extraction and evaluation [26,27,28].

One example of the attribute-level side-information is that *an attribute $a_i$ is more important than another attribute $a_j$*. These kinds of preference information are prevalent and usually much easier to obtain than precise relative weights of the attributes [29,30,31,32]. One recent attempt in this direction was done by Sun et al. [33]. Their proposed method can incorporate attribute order preferences into prototype-based clustering, and the problem of metric learning is transformed into a convex optimization problem of finding the most suitable attribute weights [33]. However, their method can only incorporate the attribute order preferences.

In this paper, we aim to address the metric learning by utilizing both the instance-level side information and the attribute-level side information. Our approach is to obtain the distance metric through an optimization method with the available side information. The proposed framework can properly generalize both kinds of side-information to similarity measures, so that these measures can be used with any of the known clustering algorithms to discover a "good" clustering that conforms to both the known facts and the user's preferences. Experimental results on a range of benchmark datasets indicate the effectiveness and the potential of the proposed approach.

The rest of the article is organized as follows. Section 2 provides a brief description of the clustering learning with instance-level and attribute-level side information respectively. Section 3 describes our learning algorithm with the combination constraints. Section 4 evaluates the proposed method on all the UCI datasets used in existing metric learning papers. In the end, we conclude the paper in Section 5.

## 2. Related Works

In this section, we briefly describe the concepts of clustering with users' constraints [17,33].

### 2.1. Metric Learning

Given a set of $n$ points $\mathscr{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ in some space of dimensionality $d$, where $\mathbf{x}_i = [x_{i1}, \cdots, x_{id}]^t$ ($t$ denotes the transpose), $\mathbf{x}_i \in \Re^d$, and the desired number of clusters $k$, the objective of clustering

is to obtain a partition of $\mathscr{X}$. While the metric learning is to obtain a vector of attribute weights $\mathbf{w} = [w_1, \cdots, w_d]^t$ ($w_i$ represents the degree of participation of attribute $i$ to the cluster), such that the instances in the same cluster are close to each other according to the $L_2$ norm distance weighted using $\mathbf{w}$, and the constraints specified by users can be satisfied.

### 2.2. *Clustering with Instance-level Side Information*

The instance-level side information usually specifies whether a pair of data instances belong to the same cluster or not. It has become one common form to represent users' prior knowledge on the application domain. For example, in information retrieval and text mining community, the rapid increasing amount of unstructured data renders it impractical to obtain individual class labels for each instance. However, it is much easier for the domain expert to provide feedback in the form of pair-wise constraints such as whether two instances belong to the same cluster or not [34,35,36].

**Definition 1. Pair-wise Constraints**[17]: for the point pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathscr{X}$, the "must-link" set $\mathscr{S}$ and "cannot-link" set $\mathscr{D}$ constraints supplied by the users can be defined as follows:

- IF $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same cluster, then $\mathbf{x}_i$ and $\mathbf{x}_j$ satisfy the "must-link" constraint.
- IF $\mathbf{x}_i$ and $\mathbf{x}_j$ are in different clusters, then $\mathbf{x}_i$ and $\mathbf{x}_j$ satisfy the "cannot-link" constraint.

In order to satisfy the constraints mentioned in Definition 1, Xing et al. [17] constructed the following optimization problem with respect to the Mahalanobis norm.

$$\min_{A} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_A^2$$

$$\textit{subject to:} \quad \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{D}} \|\mathbf{x}_i - \mathbf{x}_j\|_A \geqslant 1$$

$$A \succeq 0 \tag{1}$$

Through solving this convex optimization, Xing et al. [17] made use of the learned matrix $A$ to obtain the rescaling data instances for $\mathbf{x}_i$ with $A^{\frac{1}{2}}\mathbf{x}_i$.

$$d(\mathbf{x},\mathbf{y}) = d_A(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\| = \sqrt{(\mathbf{x}-\mathbf{y})^t A(\mathbf{x}-\mathbf{y})} \tag{2}$$

When $A$ is diagonal, the Equation (2) can be transformed into the Euclidean distance. Xing et al. [17] computed the corresponding Equation (3) of this optimization problem with Newton-Raphson technique.

$$g(A) = g(A_{11}, \ldots, A_{dd}) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_A^2 - log\Big(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{D}} \|\mathbf{x}_i - \mathbf{x}_j\|_A\Big) \tag{3}$$

### 2.3. *Clustering with Attribute-Level Side Information*

The attribute-level side information usually specifies the relative importance of a pair of attributes. In many applications, it is usually much easier to obtain this kind of attribute-level order information than the specification of the attribute weights. For example, the attribute(feature) analysis technique is broadly used in text mining, with the help of background information, important attribute(feature) can be identified [25], such as terms in title are more important than others in the content.

**Definition 2.** Attribute Order Preferences $\mathscr{P}$ represents the set satisfying attribute order preferences relationship, $p_i = (s_i, t_i, \delta_i)$ $(i = 1, 2, \cdots, m)$, $\mathscr{P} = \{p_i\}_{i=1}^m$.

Attribute Order Preferences represent the difference between the importances of two attributes. $(s,t,\delta)$ $(\delta > 0)$ means that attribute $s$ is more important than attribute $t$. Meanwhile, $(s,t,-\varepsilon)$ and $(t,s,-\varepsilon)$ ($\varepsilon$ is a small positive constant) denotes that attribute $s$ is with a similar importance as $t$.

The research of utilizing this kind of side information is still very limited. One recent attempt in this direction was done by Sun et al. [33]. They made use of attribute order preferences to construct the following optimization problem. Through an iterative updating procedure similar to the EM algorithm,

a satisfactory set of weights for attributes can be obtained.

$$\min_{\{\mathbf{w},\xi\},\{\pi_c\}_{c=1}^k,\{\mu_c\}_{c=1}^k} \frac{1}{n}\sum_{c=1}^{k}\sum_{\mathbf{x}_i\in\pi_c} D_{\mathbf{w}}(\mathbf{x}_i,\mu_c)+$$

$$\lambda_1 \sum_{p\in\mathscr{P}} \xi_p - \lambda_2 \widehat{H}(\mathbf{w})$$

s.t.

$$\mathbf{w} \in \Delta d$$

$$w_s - w_t \geqslant \delta - \xi_p \quad for\ all \quad p=(s,t,\delta)\in\mathscr{P}$$

$$\xi_p \geqslant 0 \quad for\ all \quad p\in\mathscr{P} \tag{4}$$

where $\Delta d = \{\mathbf{w}\in\Re_+^d | \mathbf{w}^t\cdot\mathbf{1}_d = 1\}$, $\Re_+^d$ denotes the set of nonnegative real number, and $\mathbf{1}_d = [\underbrace{1,\cdots,1}_{d}]^t$, $\xi = [\xi_p]$ where $p\in\mathscr{P}$.

In the function, the first term is intra-cluster distortion of the clusters $\{\pi_c\}_{c=1}^k$, which is an objective clustering validation index, $\mu_c$ is a cluster representative for each cluster $\pi_c$; the second term reflects the penalty on the constraints of attribute order preferences, which represents the attribute-level subjective criteria; the third term is a regularization term $-\widehat{H}(\mathbf{w})$ which guarantees the consistence of attribute weight.

## 3. Clustering with Combined Constraints

When both the attribute-level and the instance-level side information are present, the challenge in metric learning is how to utilize both of them into clustering process. In this section, we propose a novel learning framework which can incorporate both the pair-wise constraints and the attribute order preferences. The mathematical representation of the method are also presented in this section.

### 3.1. Learning Framework

Our aim is to generate robust and stable solutions via an optimization method considering instance-level and attribute-level side information simultaneously. In order to achieve this objective, we construct an optimization problem like this,

$$\min P_{instance-level} + \lambda P_{attribute-level}$$

where the first term, $P_{instance-level}$ denotes the penalty term of instance-level constraints, the second $P_{attribute-level}$ denotes the penalty term of attribute-level constraints, the less the value, the better the satisfactory level. The parameter $\lambda$ controls the relative contributions from each kind of side information. Through minimizing the overall objective, we can obtain the optimal attribute weights based on the constraints for clustering.

First, we follow the Xing's method, and make $A$ as diagonal to simplify the distance function so that both constraints can be incorporated easily. When $A$ is a diagonal matrix, $A = diag(A_{11},A_{22},\cdots,A_{dd}) = diag(w_1^2,\cdots,w_d^2)$, each item in the diagonal corresponds to an attribute weight $w_i$. Thus, the Mahalanobis distance between two data instances $\mathbf{x}_i$ and $\mathbf{x}_j$ can be transformed into:

$$\|\mathbf{x}_i-\mathbf{x}_j\|_A^2 = (\sqrt{(\mathbf{x}_i-\mathbf{x}_j)^t A(\mathbf{x}_i-\mathbf{x}_j)})^2 =$$

$$\mathbf{w}^t\cdot Distance(\mathbf{x}_i,\mathbf{x}_j)\cdot\mathbf{w} \tag{5}$$

in which $Distance(\mathbf{x}_i,\mathbf{x}_j)$ is a diagonal matrix, the items in the diagonal correspond to the $\mathbf{x}'_{\mathbf{ij}} = [x'_{ij1},\cdots,x'_{ijd}]$ one by one. We have $\mathbf{x}'_{\mathbf{ij}} = diag(Distance(\mathbf{x}_i,\mathbf{x}_j))$, and $x'_{ijk} = (x_{ik}-x_{jk})^2 (1\leqslant k\leqslant d)$. In this way, the instance-level side information $P_{instance-level}$ can be defined as the following optimization objective:

$$\min_{\mathbf{w}} \mathbf{w}^t\cdot Distance_S\cdot\mathbf{w}$$

s.t.

$$Distance_D\cdot\mathbf{w}\geqslant 1,$$

$$\mathbf{w}\geqslant 0 \tag{6}$$

where,

$$Distance_S = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{S}} Distance(\mathbf{x}_i,\mathbf{x}_j)$$

$$Distance_D = diag(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{D}} Distance(\mathbf{x}_i,\mathbf{x}_j)^{\frac{1}{2}}).$$

$Distance_D$ is a row vector composed with the diagonal items in matrix. The constraint $Distance_D\cdot\mathbf{w}\geqslant$

1 is added to enforce that $A$ will not collapse the dataset into a single point [17].

For the attribute-level side information in the form of attribute order preferences, we introduce a shifted hinge function [32] in the penalty term as used in Sun et al. [33]: for $p = (s,t,\delta) \in \mathscr{P}$, the penalty term for $p$ is $\xi_p = max(\delta - (w_s - w_t), 0)$.

Both the $P_{attribute-level}$ and the $P_{instance-level}$ objectives can be combined together with the control parameter $\lambda$, accordingly we can obtain the overall constrained optimization problem:

$$\min_{\mathbf{w},\xi} \mathbf{w}^t \cdot Distance_S \cdot \mathbf{w} + \lambda \sum_{p \in \mathscr{P}} \xi_p$$

s.t.

$$\begin{gathered} \mathbf{w} \in \Delta d \\ w_s - w_t \geqslant \delta - \xi_p, \forall p = (s,t,\delta) \in \mathscr{P} \\ \xi_p \geqslant 0, \forall p = (s,t,\delta) \in \mathscr{P} \\ Distance_D \cdot \mathbf{w} \geqslant 1 \end{gathered} \tag{7}$$

where $\Delta d = \{\mathbf{w} \in \Re_+^d | \mathbf{w}^t \cdot \mathbf{1}_d = 1\}$, $\mathbf{1}_d = [\underbrace{1, \cdots, 1}_{d}]^t$.

This is a linear constrained convex quadratic optimization [37], and it can be solved using many different methods, such as active set method, wolfe algorithm, Lemke algorithm, cutting plane method, *etc*.

### *3.2. Simplifying the Optimization*

For the convenience of solving the optimization problem in Equation (7) , in this section, we introduce some mathematical transformations to simplify the computation.

First, for the unification of variables in the optimization problem, we set

$$\mathbf{Y} = [w_1, \cdots, w_d, \xi_1, \cdots, \xi_m]^t$$

Furthermore, in order to transform the constraints $w_s - w_t \geqslant \delta - \xi_p$ to the form related with $\mathbf{Y}$, we define $m$ auxiliary vectors $\mathbf{a} = \mathbf{a}_p (p \in \mathscr{P})$, which describes each attribute order preference information in the set $\mathscr{P}$. For an arbitrary $\mathbf{a}_p$, we set $\mathbf{a}_p \cdot \mathbf{Y}^t = w_s - w_t + \xi_p$, and $w_s - w_t \geqslant \delta - \xi_p$ transforms into $\mathbf{a}_p \cdot \mathbf{Y}^t \geqslant \delta_p$. So, it is easy to obtain one $1 * (d+m)$ vector, $\mathbf{a}_p$ with the binary value 0 or 1. For example, if a dataset has four attributes, and $m = 1$ (the attribute preferences set $P$ contains only one preference), then $\mathbf{a}_1 = [0,-1,1,0,1]$, $\mathbf{a}_1 \cdot Y^t \geqslant \delta_1$ corresponds to $w_3 - w_2 \geqslant \delta_1 - \xi_1$, this expresses that the weight $w_3$ for the 3rd attribute is expected to be higher than the weight $w_2$ for the 2nd attribute.

Using this representation, the original optimization problem (7) can be simplified into a linear constrained convex quadratic optimization as follows:

$$\min_{\mathbf{Y}} \mathbf{Y}^t \cdot Distance_S^{'} \cdot \mathbf{Y} + \lambda \cdot \mathbf{A} \cdot \mathbf{Y}$$

s.t.

$$\begin{gathered} \mathbf{B} \cdot \mathbf{Y} = 1 \\ \mathbf{Y} \geqslant 0 \\ Distance_D \cdot C \cdot \mathbf{Y} \geqslant 1 \\ \mathbf{a}_1 \cdot \mathbf{Y} \geqslant \delta_1 \\ \vdots \\ \mathbf{a}_m \cdot \mathbf{Y} \geqslant \delta_m \end{gathered} \tag{8}$$

Here, $Distance_S^{'}$=

$$\begin{pmatrix} Distance_S & 0 \ldots 0 \\ 0 \ldots\ldots\ldots\ldots\ldots 0 \\ \vdots \ldots\ldots\ldots\ldots\ldots \vdots \\ 0 \ldots\ldots\ldots\ldots\ldots 0 \end{pmatrix}_{(d+m)\times(d+m)}$$

.

$\mathbf{A}$=$[\underbrace{0,0\cdots,0}_{d},\underbrace{1,1\cdots,1}_{m}]$,

$\mathbf{B}$=$[\underbrace{1,1\cdots,1}_{d},\underbrace{0,0\cdots,0}_{m}]$.

$C^t$=

$$\begin{pmatrix} \begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots 1 \end{pmatrix}_{d\times d} \\ \begin{pmatrix} 0 \cdots\cdots\cdots 0 \\ \cdots\cdots\cdots\cdots \\ 0 \cdots\cdots\cdots 0 \end{pmatrix}_{m\times d} \end{pmatrix}$$

Through the mathematical transformations mentioned above, the original constrained optimization problem can be transformed into a clear and simple linear constrained convex quadratic optimization problem, which can be easily solved by a range of methods.

In addition, for high-dimensional datasets, the attribute order preferences used in Sun et al.'s method are usually very small. This makes that the penalty of violating this constraint is tiny, which makes it difficult to obtain the attribute weight through optimization effectively. $(s,t,\delta)(\delta > 0)$ means that attribute $s$ is more important than $t$, the priority of $s$ is higher than $t$. Thus, we can enlarge the attribute order preferences to make $s$ more important than $t$ clearly. So, we add an appropriate parameter $\alpha$ $(0 < \alpha < 1)$ on each attribute order preference. Then, attribute order preference $(s,t,\delta)$ can be enlarged to $(s,t,\frac{\delta}{\alpha})$. For example, if $\alpha = 0.1$ and $\delta = 0.01$, it can be enlarged to $\frac{\delta}{\alpha} = \frac{0.01}{0.1} = 0.1$, and 0.1 means a large value to high-dimensional datasets.

$$\min_{\mathbf{Y}} \mathbf{Y}^t \cdot Distance'_S \cdot \mathbf{Y} + \lambda \cdot \mathbf{A} \cdot \mathbf{Y}$$

s.t.

$$\begin{gathered} \mathbf{B} \cdot \mathbf{Y} = 1 \\ \mathbf{Y} \geqslant 0 \\ Distance_D \cdot C \cdot \mathbf{Y} \geqslant 1 \\ \alpha \cdot \mathbf{a}_1 \cdot \mathbf{Y} \geqslant \delta_1 \\ \vdots \\ \alpha \cdot \mathbf{a}_m \cdot \mathbf{Y} \geqslant \delta_m \end{gathered} \tag{9}$$

In this way, each attribute order preference can be suitably enlarged by the same proportion. In the experimental result shown in Section 4, we will analyze it through comparative experiments.

### 3.3. Parameters Adjustment for $\lambda$

As the optimization objective function contains two parts, when the value of either one is much bigger than the other one, it is necessary to use the parameter $\lambda$ to make sure that none of them will be overwhelmed by the other. If the attribute-level part $P_{attribute-level}$ is relatively too small, a larger $\lambda$ is expected. In theory, the larger the parameter $\lambda$ is, the better the attribute order preferences is respected.

A rough guideline of the choice of $\lambda$ is usually related with the value of $Distance_S$, which is usually much larger than that of $\xi_p$. This in general will render the second part of the objective function ineffective. In order to incorporate two information effectively, it is necessary for the magnitude of attribute order information part is close to that of instance level information part. In fact, it corresponds to the scale of $Distance_S$. If the magnitude of attribute-level part is the same as that of instance-level part, each one can work well for our optimization. In this paper, for the convenience of keeping the two items same importance, we set $\lambda = \frac{1}{n}\sum_{i=1}^{d} Distance_S(i)$, which represents the average value of each dimensions among the "must-link" points.

For example, we randomly select some pairwise constraints and one attribute order preference, and set different values for $\lambda$ on the dataset *Iris*. The attribute order preferences is (3, 2, 0.4844), viz. $w_3 - w_2 \geqslant 0.4844$, and $\frac{1}{n}\sum_{i=1}^{d} Distance_S(i) = 0.6525$. As shown in Table 1, with $\lambda$ increases, the penalty term of attribute order preferences gradually decreases, and the attribute order preference is extremely satisfied when $\lambda = \frac{1}{n}\sum_{i=1}^{d} Distance_S(i) = 0.6525$. However, too large value for $\lambda$ is not appropriate because it may overwhelm the instance-level information. We will show the influence of $\lambda$ values on clustering results in experimental results.

Table 1. Different values for $\lambda$ on *Iris*

| $\lambda$ | $\xi$ |
|---|---|
| 0 | 0.4844 |
| 0.1 | 0.2844 |
| 0.2 | 0.1815 |
| 0.3 | 0.0944 |
| 0.4 | 0.0072 |
| 0.5 | 8.35e-9 |
| 0.6525 | 1.00e-09 |
| 0.7 | 4.75e-10 |

### 3.4. The Main Algorithm

A large number of clustering algorithms heavily rely on the distance measure over the data instance space. Accordingly, a suitable distance defined for the application domain, will be able to be used with most clustering algorithms. The method presented in this paper can be used with any clustering algorithm

which requires a distance measure. In our experiments, we use $k$-means clustering algorithm in order to get a fair comparison with Xing et al.'s method.

With the input dataset $\mathscr{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, number of output clusters $k$, must-link constraints $\mathscr{S}$, cannot-link constraints $\mathscr{D}$, and attribute order preferences $\mathscr{P}$, parameters $\lambda$, our main algorithm includes the following steps:

1. Formulate the optimization problem according to the Equation (9);
2. Solve the optimization to obtain the weights $\mathbf{w}$.
3. Rescale each point in original datasets with learned attribute weights $\mathbf{w}$.
4. Clustering datasets transformed with learned attribute weights $\mathbf{w}$.

As the MOSEK package* can work fast and effectively, we use it to solve the optimization problem with combination constraints in this paper. Time complexity of our optimization solved by MOSEK is $O(\sqrt{n}log(1/\varepsilon))$ [38], and it costs only about 1 second even on large dataset with lots of side information.

## 4. Experimental Evaluation

In this section, we provide empirical evidence for the validity of our method through a comprehensive set of experiments. We first describe the datasets used and experimental settings in section 4.1. Then in section 4.2, we introduce the evaluation criteria in this paper for assessing the algorithm performance. Followed by the results comparisons in section 4.3, we demonstrate the comparative performance of the proposed method with $k$-means, Xing's method [17] and MPCK-Means [15]. For a fair comparison, we randomly select initial centroids for MPCK-Means, and make sure cluster centroids of the three methods are the same.

*http://www.mosek.com

†ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data/

### 4.1. Datasets and Experimental Setting

For performance evaluation, this paper used the datasets drawn from UCI machine learning repository†. Table 2 summarizes the characteristics of these datasets. As indicated in Table 2, these datasets vary significantly in their sizes, number of clusters, and number of attributes.

In the experiments, for the high dimensional dataset *Ionosphere* and *Spam*, we set magnification factor $\alpha = 0.1$ (described as section 3.2).

In our experiment, for convenience, we generate simulated attribute order preferences by using the ground truth class information similar as Sun et al. [33]. Firstly, calculating the within-class distortion $\Theta_j = \frac{1}{v_j}\sum_{c=1}^{k}\sum_{\mathbf{x}_i \text{ in class } c}(x_{ij}-\mu_{cj})^2$ for each dimension $1 \leqslant j \leqslant d$, where $v_j = \frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\frac{1}{n}\sum_{i=1}^{n}x_{ij})^2$, then, calculating the inverse within-class distortion $\Gamma_j = \frac{\sum_{l\neq j}\Theta_l}{\Theta_j}$. After that, estimating the optimal feature weights by $\widetilde{w_j} = \frac{\Gamma_j}{\sum_{l=1}^{d}\Gamma_l}$, which is just a rough estimate of the optimal attribute weighting.

Table 2. Dataset characteristics

| Dataset | #Examples($n$) | #Attribute($d$) | #Clusters($k$) |
|---|---|---|---|
| Iris | 150 | 4 | 3 |
| Diabetes | 768 | 8 | 2 |
| Soybean | 47 | 35 | 4 |
| Wdbc | 569 | 30 | 2 |
| Spam | 4601 | 57 | 2 |
| Ionosphere | 351 | 34 | 2 |
| Protein | 116 | 20 | 6 |
| Balance | 625 | 4 | 3 |

Based on the estimated attribute weight vector $\widetilde{\mathbf{w}}$, the largest and smallest $\lfloor\frac{d}{2}\rfloor$ attribute weight can be obtained. For the attribute order, we select $\lfloor\frac{d}{4}\rfloor$ pairs with the largest $\delta$ values as the attribute level information. For datasets *Ionosphere* and *Soybean*, there are some column attributes whose value is all same, this results in that we cannot generate attribute order preferences with this method. For the dataset *Ionosphere*, the attribute order preferences can be obtained after deleting the second attribute; for the dataset *Soybean*, after deleting 14 attributes with the same value, we still need add one tiny positive real

(such as 0.0001) for avoiding the 0 value of denominator.

For the dataset *Protein*, one attribute may be more important, which incur that the attribute order preferences are related with it. Thus, we use another method as the following. For the obtained attribute weights, we select the attributes with the largest and smallest weight as a pair, then delete them, continue another pair for the final results.

### 4.2. Evaluation Criteria

The notion of cluster validation refers to the quantitative and objective evaluation of the output of a clustering algorithm. Evaluating the quality of clustering is a fundamental problem in unsupervised learning. In the absence of prior information, it is in general a difficult task, and there are usually three different criteria: internal, relative, and external [39].

Typically, clustering results are evaluated using the external indices for measuring how similar a clustering is to another clustering, through assessing the performance by matching cluster structure to a predefined reference ground truth, such as Jaccard, Rand, F-Measure, NMI, *etc*. Since we assess the performance and validity of our algorithm with Xing et al. [17], we use the same evaluation criteria as in [17]. Additionally, for more complete performance evaluation, we use the normalized mutual information (NMI) and pairwise F-Measure as other measurements.

#### 4.2.1. Rand Statistic (RS)

As a pair counting approach, rand statistic [40] measures the degree of correspondence between a pre-specified structure and the clustering results to data points $X$. Using a generalized $2\times2$ contingency matrix, it judges the percentage of member pairs two clustering have in common for performance evaluation.

Let $C = \{c_1, \cdots, c_r\}$ denote the clustering result in the dataset $X$, $P = \{p_1, \cdots, p_k\}$ presents the clusters. Then, referring to $x \in X$, $y \in X$, we have the following terms.

- **SS**: If $x \in c_i$, $y \in c_i$, and $x \in p_j$, $y \in p_j$, then $(x,y) \in SS$.
- **SD**: If $x \in c_i$, $y \in c_i$, and $x \in p_j$, $y \in p_t (t \neq j)$, then $(x,y) \in SD$.
- **DS**: If $x \in c_i$, $y \in c_t (t \neq i)$, and $x \in p_j$, $y \in p_j$, then $(x,y) \in DS$.
- **DD**: If $x \in c_i$, $y \in c_t (t \neq i)$, and $x \in p_j$, $y \in p_s (s \neq j)$, then $(x,y) \in DD$.

Let $a$, $b$, $c$, $d$ refer to number in the set $SS$, $SD$, $DS$, $DD$, and $M = n \cdot (n-1)/2$, we can define the clustering accuracy as:

$$R = (a+d)/M$$

This describes the fraction of the total number of pairs members that occur in the same cluster in both clusterings and the number of pairs of members that don't occur in the same cluster in either clusterings compared to the total number of pairs [41].

#### 4.2.2. Normalized Mutual Information (NMI)

Different from the pair method, normalized mutual information is one kind of measure based on information entropy and is utilized in many researches [42,43,44,45].

$$NMI(\mathscr{C},\mathscr{B}) = \frac{I(\mathscr{C};\mathscr{B})}{\sqrt{H(\mathscr{C})H(\mathscr{B})}} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} n_{ij} log \frac{n \cdot n_{ij}}{n_i \cdot n'_j}}{\sqrt{\sum_{i=1}^{k} n_i log \frac{n_i}{n} \sum_{j=1}^{k} n'_j log \frac{n'_j}{n}}}$$

Here, $H$ presents the entropy, and $I$ computes the mutual information. $\mathscr{C}$ presents the clustering results after applying our approach to $X$, and $\mathscr{B}$ denotes the pre-specified structure. The number of items in $\mathscr{C}$ and $\mathscr{B}$ are both $k$. We use $n_i$ to express the object number in the $i$th cluster, $n'_j$ denotes the one in the $j$th cluster. $n_{ij}$ denotes the item number included in $i$th and $j$th cluster.

#### 4.2.3. Pairwise F-Measure

F-measure derived from the traditional information retrieval, and utilized same-cluster pairs to evaluate clustering quality:

$$Precision = \frac{PairsCorrectlyPredictedInSameCluster}{TotalPairsPredictedInSameCluster}$$

$$Recall = \frac{PairsCorrectlyPredictedInSameCluster}{TotalPairsInSameCluster}$$

$$F-Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

### *4.3. Results Comparison*

We compare the proposed clustering framework with the methods proposed in Xing et al. [17] and MPCK-Means [15]. The involved optimization problems in this paper and Xing's work [17] are solved by the Matlab MOSEK optimization toolbox.

#### *4.3.1. Clustering Accuracy Comparison*

For each the dataset, the same largest $\lfloor \frac{d}{4} \rfloor$ weight attribute order pairs, together with a set of randomly selected pair-wise constraints (5% must-link constraints and 6% cannot-link constraints [‡]) are provided into each metric learning algorithm for the instance-level information and the attribute level information. In order to get a fair comparison, we run each algorithm 10 times, and the average measures from these 10 runs are recorded for comparative analysis. Fig.1, Fig.2 and Fig.3 show the result comparisons on the rand statistic (RS), the normalized mutual information (NMI) indexes and pairwise F-measure respectively.

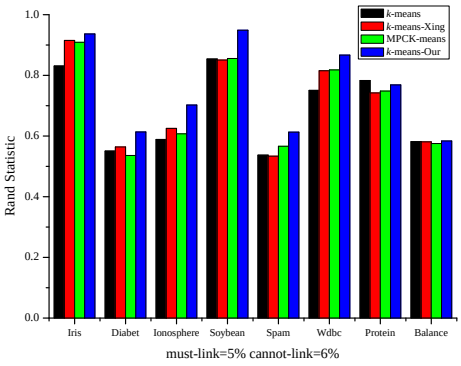



Fig. 1. The result comparison of RS

Clearly, our clustering method with instance-level and attribute-level side information are capable of achieving superior accuracy results with respect to the Xing's and Bilenko's methods on all the tested datasets. For the dataset *Balance*, because the attribute level information is $[0.25, 0.25, 0.25, 0.25]$, which means that all attributes are equally important, this actually renders the attribute-level information ineffective, and means that no improvement can be achieved by the metric learning algorithms over the regular $k$-means algorithm.

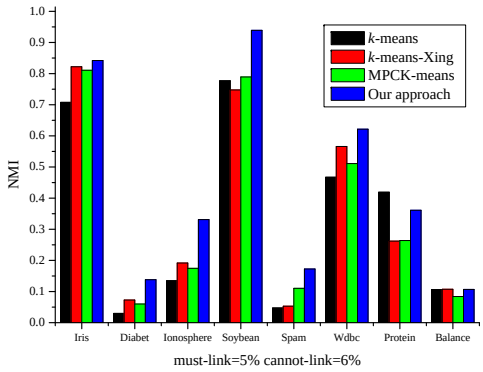



Fig. 2. The result comparison of NMI

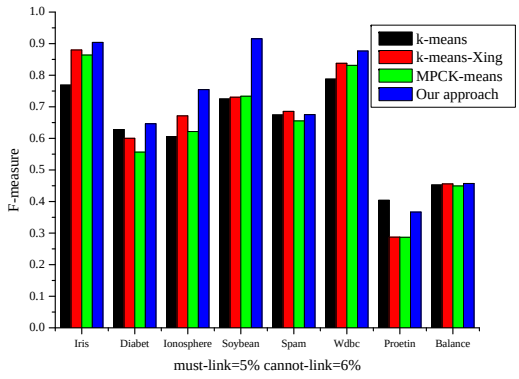



Fig. 3. The result comparison of F-Measure

Then, we also report the $t$-test evaluation for assessing the robustness of our method through measuring the statistical significance of these results. Usually, when the value is lower than 0.05, we can conclude with 95% confidence that the two methods

[‡]This means in a dataset of 100 points, we used only 5 must-link and 6 cannot-link constraints

are statistically different in performance. Table 3 - Table 5 present the paired $t$-test results respectively under RS index, NMI and F-Measure. It is obvious that in almost all cases [§], the improvement over Xings' method and MPCK-means is significant.

Also, we obtain the statistical analysis result with 80 records (10 tests in each datasets) as Table 6. This indicates that the improvement of our method is significant over the $k$-means and the Xing's and Bilenko's methods.

Table 3. $t$-test under RS index

| Our approach | $k$-means | $k$-means-Xing | MPCK-means |
|---|---|---|---|
| Iris | 5.5166e-004 | 0.3514 | 0.0055 |
| Diabetes | 5.6334e-035 | 0.0038 | 1.9713e-013 |
| Soybean | 0.0055 | 0.0200 | 0.0037 |
| Wdbc | 3.7363e-022 | 0.0048 | 9.2679e-009 |
| Spam | 2.2432e-005 | 4.3680e-005 | 0.0458 |
| Ionosphere | 9.7811e-008 | 0.0081 | 2.6172e-006 |
| Protein | 0.1584 | 0.0333 | 0.0389 |
| Balance | 0.6197 | 0.5973 | 0.0324 |

Table 4. $t$-test under NMI

| Our approach | $k$-means | $k$-means-Xing | MPCK-means |
|---|---|---|---|
| Iris | 9.2492e-005 | 0.4074 | 0.0325 |
| Diabetes | 3.5817e-028 | 0.0189 | 1.5387e-012 |
| Soybean | 4.7518e-004 | 6.7157e-004 | 4.9228e-004 |
| Wdbc | 2.8513e-016 | 0.0238 | 6.0432e-009 |
| Spam | 1.0554e-006 | 2.6554e-004 | 0.0081 |
| Ionosphere | 7.6949e-012 | 0.0033 | 2.8673e-009 |
| Protein | 0.1152 | 0.0189 | 0.0075 |
| Balance | 0.9337 | 0.9469 | 0.0011 |

Table 5. $t$-test under F-measure

| Our approach | $k$-means | $k$-means-Xing | MPCK-means |
|---|---|---|---|
| Iris | 1.0031e-004 | 0.3653 | 0.0064 |
| Diabetes | 2.6989e-011 | 7.1729e-004 | 2.1465e-014 |
| Soybean | 0.0016 | 0.0074 | 0.0015 |
| Wdbc | 3.4647e-020 | 0.0051 | 1.0708e-008 |
| Spam | 0.8672 | 0.0669 | 0.4960 |
| Ionosphere | 2.0308e-015 | 0.0038 | 1.0273e-012 |
| Protein | 0.1539 | 0.0113 | 0.0041 |
| Balance | 0.4138 | 0.7807 | 0.0753 |

Table 6. $t$-test with 80 records

| Our approach | $k$-means | $k$-means-Xing | MPCK-means |
|---|---|---|---|
| RS | 3.6329e-014 | 5.7858e-008 | 3.8536e-007 |
| NMI | 1.2980e-012 | 4.2926e-009 | 3.5164e-008 |
| F-measure | 2.4974e-007 | 8.3019e-006 | 1.1448e-010 |

### 4.3.2. *Clustering Accuracy versus Constraints*

Fig.4 - Fig.7 show the effect of the size of pair-wise constraints on the quality of clustering on four UCI datasets (Diabet, Wdbc, Ionosphere and Protein). Most of these datasets are selected as "difficult-to-cluster" by Halkidi [19].

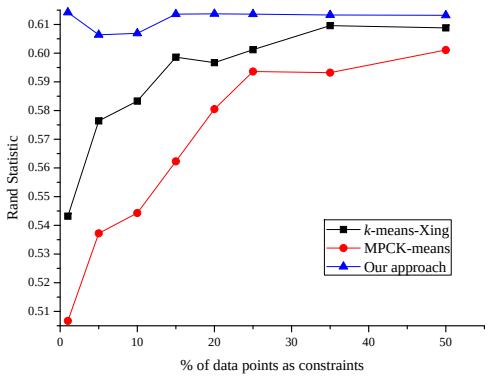



Fig. 4. Clustering accuracy versus constraints on Diabet

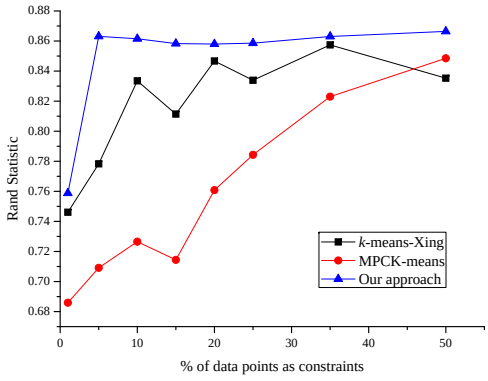



Fig. 5. Clustering accuracy versus constraints on Wdbc

[§]except the *Balance* data set, on which the attribute-level information does not work effectively

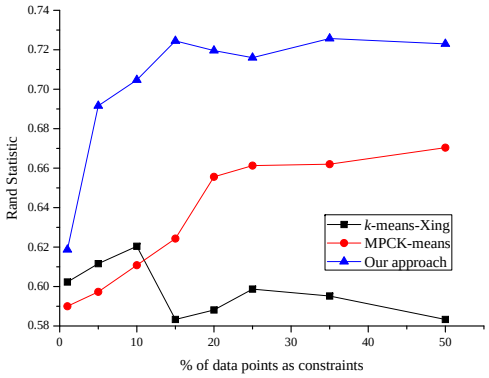



Fig. 6. Clustering accuracy versus constraints on Ionosphere

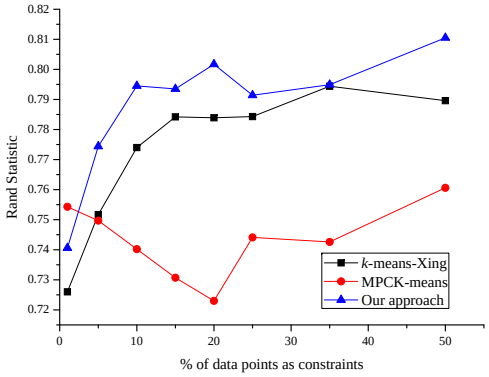



Fig. 7. Clustering accuracy versus constraints on Protein

The curves corresponding to *k*-means-Xing, MPCK-means and our approach have slightly different variations. We observe that our approach systematically leads to improvement in clustering quality and fast converge even in cases where few pairwise constraints are used. For the *Wdbc* dataset, our clustering algorithm can obtain an abrupt initial increase up using only 5% of the data points as constraints, while Xing's method and MPCK-means cannot obtain this accuracy even with 50% constraints. In the dataset *Diabetes* and *Ionosphere*, our method can obtain a better convergent result than the others. Especially for the dataset *Diabetes*, our method can obtain 61.369% accuracy only with 1% constraints, whereas additional constraints improve the clustering accuracy only insubstantially. A possible explanation for this is the distribution of the underlying data. In addition, our method can steadily increase clustering quality with more and more constraints, while Xing's method decreases on *Ionosphere* dataset and MPCK-means decreases on *Protein* dataset. Table 7 also prove robustness of our method. The decrease of Xing's method and MPCK-means is mainly due to low coherence of pairwise constraints [7], while our method can still effectively work by attribute order preferences.

Based on above analysis, we can conclude with empirical evidence that our learning approach significantly outperforms Xing's method and MPCK-means, which can only utilize the instance level information.

Table 7. *t*-test on the four difficult datasets with all trial results

| Our approach | *k*-means-Xing | MPCK-means |
|---|---|---|
| Diabetes | 0.0233 | 0.0045 |
| Wdbc | 0.0116 | 0.0194 |
| Ionosphere | 2.2745e-004 | 1.0716e-004 |
| Protein | 0.0014 | 0.0028 |

#### *4.3.3. The Ratio between Must-link and Cannot-link*

Setting a proper ratio between must-link and cannot-link is one problems which is usually considered in utilizing metric learning, however, most of existing work didn't consider this ratio [5,12,15,17] except Halkidi [19].

Fig.8 shows clustering results with 10% randomly selected must-link or cannot-link constraints without setting a ratio between must-link and cannot-link.

From the Figure, it can be seen that the performance of our method is in general better than Xing's method and MPCK-means on most of datasets.

In the other part of this paper, for the convenience of comparison, we use 5% must-link and 6% cannot-link according to the paper [19].

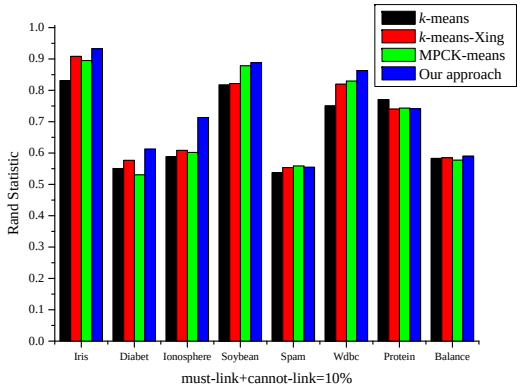


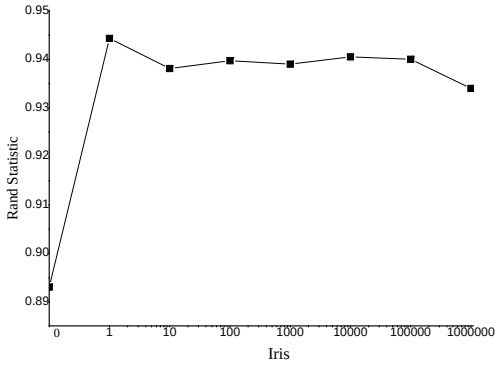

Fig. 8. Clustering results with 10% randomly selected constraints (This means in a dataset of 100 points, we randomly select 10 constraints without setting ratio between must-link and cannot-link)

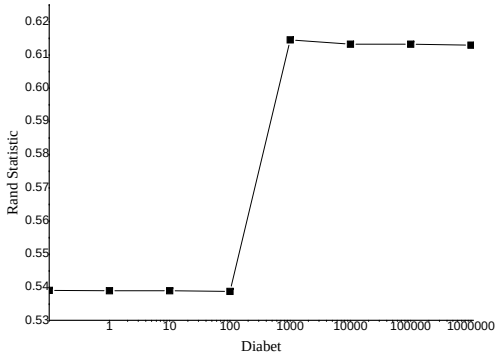
### 4.3.4. *Setting the Parameter* $\lambda$

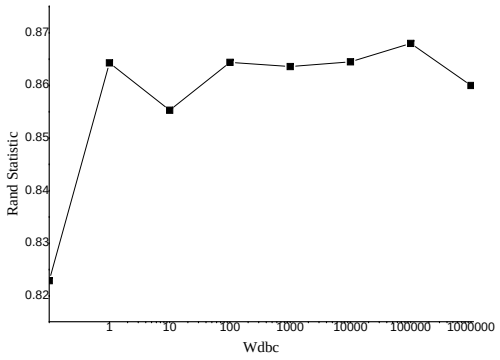
As shown in Fig.9, clustering quality improves with $\lambda$ value increases provided that $\lambda$ value is not very large. When $\lambda$ value is too large, the weight of attribute-level information becomes too large comparing to instance-level information. It may overwhelm instance-level information and have negative on clustering result. For example, according to Fig.9 and Table 8, *Iris* and *Ionosphere* can get best result when $\lambda = \frac{1}{n}\sum_{i=1}^{d} Distance_S(i)$, and clustering quality decreases with larger $\lambda$ value. Although the result is not perfect on some datasets, it is appropriate to set $\lambda = \frac{1}{n}\sum_{i=1}^{d} Distance_S(i)$ on the whole datasets, so as to avoid exhaustedly searching appropriate value for $\lambda$ and treat both kinds of information as equally as possible.

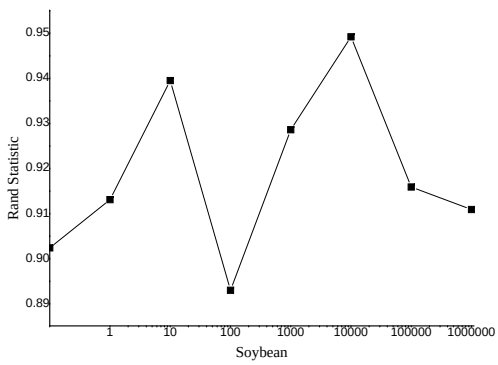
Table 8. set $\frac{1}{n}\sum_{i=1}^{d} Distance_S(i)$ for $\lambda$

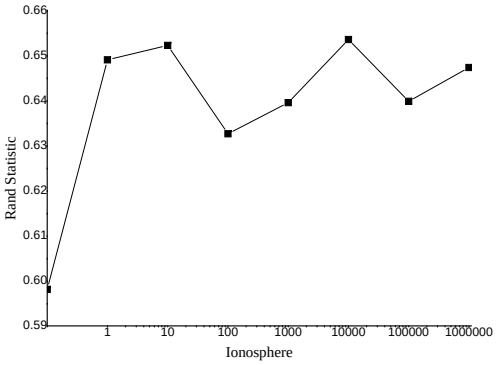
| Dataset | $\lambda$ value | clustering result (RS) |
|---|---|---|
| Iris | 2.3075 | 0.9432 |
| Diabetes | 8.9921e+004 | 0.6123 |
| Wdbc | 1.4823e+005 | 0.8606 |
| Soybean | 0.8095 | 0.9187 |
| Spam | 3.1683e+006 | 0.5477 |
| Protein | 72.0500 | 0.7535 |
| Ionosphere | 8.0863 | 0.6732 |
| Balance | 86 | 0.5829 |

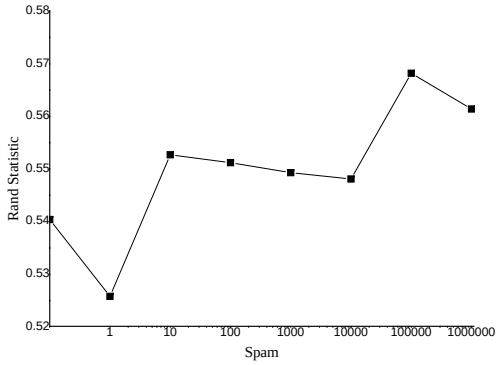


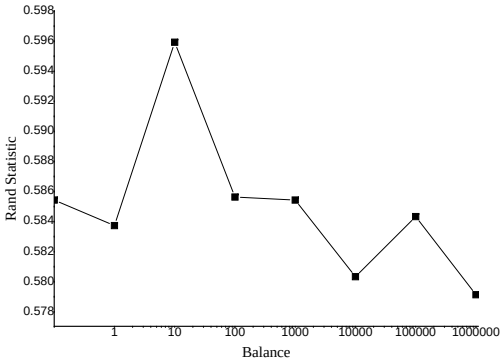





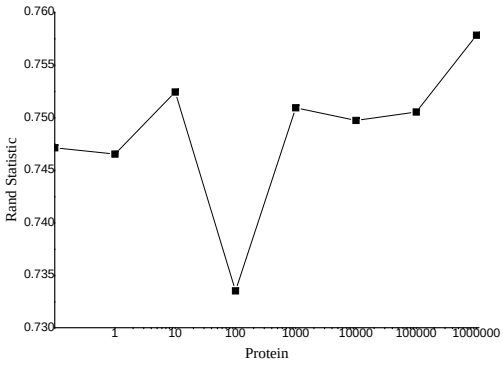




Fig. 9. clustering results (RS) with different values to set $\lambda$

### *4.3.5. Time Complexity Evaluation*

According to Equation (7), the efficiency of MOSEK toolbox is independent of numbers of pairwise constraints, and may be affected by dimension of $\mathbf{w}$ (the dimension of dataset) and number of attribute order preferences. Thus, we have experiments for the two potential factors. As shown in Fig.10 and Fig.11, time complexity of our optimization is very low, close to 1 second. This demonstrates that the two factors have no influences on time complexity in our optimization process.

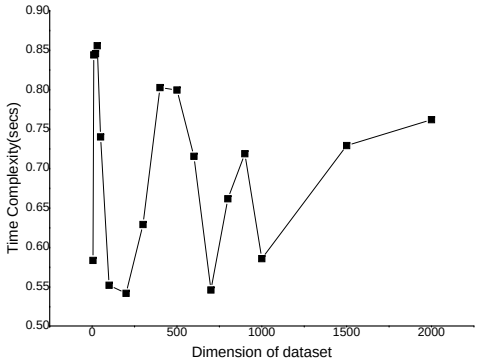



Fig. 10. Time Complexity of our optimization versus dimension of dataset

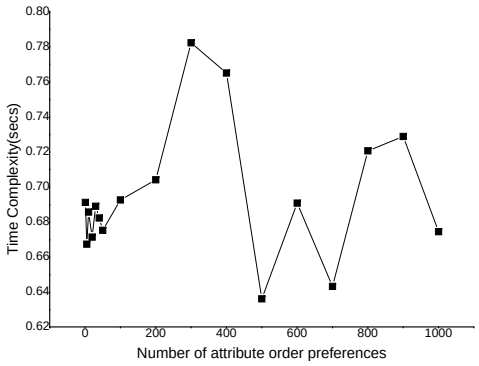



Fig. 11. Time Complexity of our optimization versus number of attribute order preferences

## 5. Conclusions

In this paper, we have studied the problem of improving data clustering by using both instance and attribute level side information. Up to our knowledge, this is one of the first attempt to systematically incorporate these two kinds of side information together in the unsupervised learning context. A general learning framework simultaneously combining these two kinds of information, in the form of pairwise constraints and the attribute order preferences respectively, has been proposed to learn a new distance metric. The problem is then transformed into an optimization task which can be solved by a wide range of existing mathematical methods. Additionally, we have analyzed the detail factors in learning framework and the parameter setting for better performance. Experimental results show that our method can work effectively and significantly has improved the clustering performance compared with algorithms using the pair-wise constraints only.

In the future, we plan to incorporate the objective criterion to further improve the cluster quality, and apply our method to some specific applications.

*Acknowledgement*

We thank the anonymous reviewers for their detailed and extremely helpful comments.

This work was partially supported by the National Natural Science Foundation of P.R.China (No.60802066), the China Postdoctoral Science Foundation (No.20100471494), the Excellent Young Scientist Foundation of Shandong Province of China under Grant (No.2008BS01009), the Science and Technology Planning Project of Shandong Provincial Education Department (No.J08LJ22).

## References

1. A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: a review", *ACM Computing Surveys*, **31**(3):264-323(1999).
2. R. K. Brouwer, "Clustering feature vectors with mixed numerical and categorical attributes", *International Journal of Computational Intelligence Systems*, **1**(4):285-298(2008).
3. R. K. Brouwer, "Fuzzy relational fixed point clustering", *International Journal of Computational Intelligence Systems*, **2**(1):69-82(2009).
4. S. Ilhan, N. Duru and E. Adali, "Improved fuzzy art method for initializing k-means", *International Journal of Computational Intelligence Systems*, **3**(3):274-279(2010).
5. S. Basu, M. Bilenko and R. J. Mooney, "A probabilistic framework for semi-supervised clustering", *Proc. of the 10th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 59-68(2004).
6. N. Grira, M. Crucianu and N. Boujema, "Unsupervised and semi-supervised clustering: a brief survey", *In a Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (FP6)*(2005).
7. I. Davidson, K. Wagstaff and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms", *Proc. of the 10th Euro. Conf. on Principle*

*and Practice of Knowledge Discovery in Databases*, 115-126(2006).
8. L. Yang and R. Jin, "Distance metric learning: A comprehensive survey", *Michigan State Universiy*, (2006).
9. R. Kulis, S. Basu, I. Dhillon and R. Mooney, "Semi-supervised graph clustering: a kernel approach", *Mach. Learn.*, **74**:1-22(2009).
10. X. S. Yin, S. C. Chen, E. L. Hu and D. Q. Zhang, "Semi-supervised clustering with metric learning: an adaptive kernel method", *Pattern Recognition*, **43**(4):1320-1333(2010).
11. K. Wagstaff and C. Cardie, "Clustering with instance-level constraints", *Proc. of the 17th Intl. Conf. on Machine Learning*, 1103-1110(2000).
12. K. Wagstaff, C. Cardie, S. Rogers and S. Schrödl, "Constrained k-means clustering with background knowledge", *Proc. of the 18th Intl. Conf. on Machine Learning*, 577-584(2001).
13. D. Klein, S. D. Kamvar and C. D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering", *Proc. of the 19th Intl. Conf. on Machine Learning*, 307-314(2002).
14. N. Shental, A. Bar-hillel and D. Weinshall, "Computing gaussian mixture models with em using equivalence constraints", *Advances in Neural Information Processing Systems 16*, (2003).
15. M. Bilenko, S. Basu and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering", *Proc. of the 21st Intl. Conf. on Machine Learning*, 81-88(2004).
16. N. Kumar and K. Kummamuru, "Semi-supervised clustering with metric learning using relative comparisons", *IEEE Transactions on Knowledge and Data Engineering*, **20**(4):496-503(2008).
17. E. P. Xing, A. Y. Ng, M. I. Jordan and S. J. Russell, "Distance metric learning with application to clustering with side-information", *Advances in Neural Information Processing Systems 15*, 505-512(2002).
18. A. Bar-Hillel, T. Hertz, N. Shental and D. Weinshall, "Learning a mahalanobis metric from equivalence constraints", *J. Mach. Learn. Res.*, **6**:937-965(2005).
19. M. Halkidi, D. Gunopulos, M. Vazirgiannis, N. Kumar and C. Domeniconi, "A clustering framework based on subjective and objective validity criteria", *ACM Trans. Knowl. Discov. Data.*, **1**(4):1-25(2008).
20. S. Xiang, F. Nie and C. Zhang, "Learning a Mahalanobis distance metric for data clustering and classification", *Pattern Recognition*, **41**(12):3600-3612(2008).
21. S. Basu, A. Banerjee and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering", *Proc. of the 4th SIAM Intl. Conf. on Data Mining*, 333-344(2004).
22. A. Huang, D. Milne, E. Frank and I. H. Witten, "Clustering documents with active learning using Wikipedia", *Proc. of the 8th IEEE Intl. Conf. on Data Mining*, 839-844(2008).
23. R. Huang and W. Lam, "An active learning framework for semi-supervised document clustering with language modeling", *Data & Knowledge Engineering*, **68**(1):49-67(2009).
24. J. Wang, S. Wu, Vu. H and G. Li, "Text document clustering with metric learning", *Proc. of the 33rd Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 783-784(2010).
25. S. Banerjee, K. Ramanathan and A. Gupta, "Clustering short texts using wikipedia", *Proc. of the 30th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 787-788(2007).
26. I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin and C. G. Nevill-Manning, "KEA: Practical automatic keyphrase extraction", *Proc. of the 4th ACM Conf. on Digital Libraries*, 255(1999).
27. P. D. Turney, "Learning to extract keyphrases from text", *National Research Council, Institute for Information Technology, Technical Report ERB-1057*, (1999).
28. X. Wu and A. Bolivar, "Keyword extraction for contextual advertisement", *Proc. of the 17th Intl Conf. on World Wide Web*, 1195-1196(2008).
29. T. Joachims, "Optimizing search engines using clickthrough data", *Proc. of the 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 133-142(2002).
30. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and G. Hullender, "Learning to rank using gradient descent", *Proc. of the 22nd Intl. Conf. on Machine Learning*, 89-96(2005).
31. S. Yu, K. Yu, V. Tresp and H. P. Kriegel, "Collaborative ordinal regression", *Proc. of the 23rd Intl. Conf. on Machine learning*, 1089-1096(2006).
32. X. Zhu and A. Goldberg, "Kernel regression with order preferences", *Proc. of the 22nd AAAI Conf. on Artificial Intelligence*, 681-687(2007).
33. J. Sun, W. Zhao, J. Xue, Z. Shen and Y. Shen, "Clustering with feature order preferences", *Proc. of the 10th Pacific Rim Intl. Conf. on Artificial Intelligence*, 382-393(2008).
34. X. Ji and W. Xu, "Document clustering with prior knowledge", *Proc. of the 29th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 405-412(2006).
35. Y. Chen, M. Rege, M. Dong and J. Hua, "Incorporating user provided constraints into document clustering", *Proc. of the 7th IEEE Intl. Conf. on Data Mining*, 103-112(2007).
36. G. Hu, S. Zhou, J. Guan and X. Hu, "Towards effective document clustering: A constrained k-means based approach", *Inf. Process. Manage.*, **44**(4):1397-

1409(2008).
37. S. Boyd and L. Vandenberghe, "Convex optimization", *Cambridge University Press*, (2004).
38. E. D. Andersen and Y. Ye, "On a homogeneous algorithm for the monotone complementarity problem", *Mathematical Programming*, **84**(2):375-399(1999).
39. A. K. Jain and R. C. Dubes, "Algorithms for clustering data", *Prentice-Hall, Inc.*, (1988).
40. M. Halkidi, Y. Batistakis and M. Vazirgiannis, "On Clustering Validation Techniques", *Journal of Intelligent Information Systems*, **17**(2-3):107-145(2001).
41. D. Pfitzner, R. Leibbrandt and D. Powers, "Characterization and evaluation of similarity measures for pairs of clusterings", *Knowl. Inf. Syst.*, **19**:361-394(2009).
42. X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach", *Prof. of the 20th Intl. Conf. on Machine Learning*, 186-193(2003).
43. A. Fred and A. Jain, "Robust data clustering", *Proc. of the 2003 IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, **2**, 128-136(2003).
44. X. Yin, E. Hu and S. Chen, "Discriminative semi-supervised clustering analysis with pairwise constraints", *Journal of Software(in Chinese)*, **19**(11):2791-2802(2008).
45. X. Hu, X. Zhang, C. Lu, E. K. Park and X. Zhou, "Exploiting Wikipedia as external knowledge for document clustering", *Proceedings of the 15th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 389-396(2009).