Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

# Smart Multi-Task Bregman Clustering and Multi-Task Kernel Clustering

**Xianchao Zhang** and **Xiaotong Zhang**
School of Software Technology
Dalian University of Technology
Dalian 116620, China
xczhang@dlut.edu.cn, zhangxiaotong1022@126.com

## Abstract

Multitask Bregman Clustering (MBC) alternatively updates clusters and learns relationship between clusters of different tasks, and the two phases boost each other. However, the boosting does not always have positive effect, it may also cause negative effect. Another issue of MBC is that it cannot deal with nonlinear separable data. In this paper, we show that MBC's process of using cluster relationship to boost the updating clusters phase may cause negative effect, i.e., cluster centroid may be skewed under some conditions. We propose a smart multi-task Bregman clustering (S-MBC) algorithm which identifies negative effect of the boosting and avoids the negative effect if it occurs. We then extend the framework of S-MBC to a smart multi-task kernel clustering (S-MKC) framework to deal with nonlinear separable data. We also propose a specific implementation of the framework which could be applied to any Mercer kernel. Experimental results confirm our analysis, and demonstrate the superiority of our proposed methods.

## Introduction

Clustering is a fundamental problem in machine learning and data mining. Traditional clustering methods deal with a single clustering task on a single data set. However, there are many related tasks in real applications, such as web pages from different universities and slowly time-evolving data sets, which motivate multi-task clustering. Multi-task clustering is desired to improve the clustering performance of individual tasks through learning the relationship between clusters of different tasks.

Recently, (Zhang and Zhang 2010) proposed a Bregman divergence based multi-task clustering (MBC) algorithm, which can solve the multi-task clustering problem when the data sets are from a same distribution or similar distributions (the tasks share a set of data points). MBC alternatively updates clusters and learns the relationship between clusters of different tasks, and the two phases boost each other. The alternative boosting was shown by (Zhang and Zhang 2010) to gain better performance compared with single-task Bregman divergence clustering. However, the boosting of the second phase to the first phase does not always have positive effect as desired, it may cause negative effect under some conditions. Another issue of MBC is that it cannot deal with nonlinear separable data.

In this paper, we show that using the cluster relationship to boost clustering can bring negative effect, i.e., for data points not shared by other tasks, the centroids may be skewed. We propose a smart multi-task Bregman clustering (S-MBC) algorithm which identifies the negative effect of the boosting and avoids the negative effect if it occurs. The basic idea of S-MBC is using a local loss of each task to measure whether the negative effect of boosting clusters in this task occurs, i.e., the centroids deviate from the optimal positions. If the local loss of one task calculated by MBC is larger than the single-task Bregman divergence clustering, S-MBC will stop using the boosting. To tackle the nonlinear separable data multi-task clustering problem, we introduce Mercer kernel into the framework of S-MBC and get a smart multi-task kernel clustering (S-MKC) framework. We also propose a specific optimization method, which is quite different from that of MBC, to implement the multi-task kernel clustering framework.

Experimental results confirm our observation on the negative effect of MBC's boosting under some conditions, and demonstrate the superiority of our proposed S-MBC in terms of avoiding the negative effect, and the superiority of our proposed S-MKC for nonlinear separable data.

## Related Work

(Caruana 1997), (Ando and Zhang 2005) and (Argyriou, Evgeniou, and Pontil 2006) have proposed algorithms to tackle supervised multi-task learning. Recently, multi-task clustering (unsupervised multi-task learning) attracts much attention. (Gu and Zhou 2009) learned a subspace shared by multiple related tasks. (Gu, Li, and Han 2011) handled multi-task clustering problem by learning a kernel. Both the two methods above focus on cross-domain multi-task clustering, and assume all tasks share an identical set of clusters, which is too restrictive in practice. (Zhang and Zhang 2010) proposed a multitask Bregman clustering algorithm, which alternatively updates the single-task Bregman clustering and learns the relationship between clusters of different tasks, and the two phases boost each other. It has no need to assume that all tasks share an identical set of clusters. Empirical results show that MBC can improve the clustering per-

formance over the traditional Bregman divergence clustering, but it may cause negative effect under some conditions and cannot solve multi-task clustering problem of nonlinear separable data.

## Multi-task Bregman Clustering Framework

### Problem Formulation

Suppose we are given $T$ clustering tasks, each with a set of points, i.e., $X^{(t)} = \{x_1^{(t)}, x_2^{(t)}, ...., x_{n^{(t)}}^{(t)}\}, 1 \leq t \leq T$, where $n^{(t)}$ is the number of data points in the $t$-th task. Each data corpus is to be partitioned into $c^{(t)}$ clusters. For each task $t$, we need to find a partition $P^{(t)} = \{M^{(t)}, h^{(t)}\}$, which is defined by a set of centroids $M^{(t)} = \{m_1^{(t)}, ..., m_{c^{(t)}}^{(t)}\}$ and an assigning function $h^{(t)} : X^{(t)} \rightarrow \{1, ..., c^{(t)}\}$. $P = \{P^{(t)}\}_{t=1}^{T}$ denotes all the partitions, $M = \{M^{(t)}\}_{t=1}^{T}$ denotes the set of all the centroids, and $H = \{h^{(t)}\}_{t=1}^{T}$ denotes all the assigning functions. $d_\phi(x, y)$ denotes the Bregman divergence between data $x$ and $y$.

### Multitask Bregman Clustering

(Zhang and Zhang 2010) proposed a general framework for multi-task clustering to minimize the following loss function

$$J = \frac{1}{T}\sum_{t=1}^{T} L^{(t)}(P^{(t)}, X^{(t)}) + \lambda\Omega(P). \tag{1}$$

In Eq.(1), $L^{(t)}(P^{(t)}, X^{(t)}) = \frac{1}{n^{(t)}}\sum_{i=1}^{n^{(t)}} d_\phi(x_i^{(t)}||m_{h^{(t)}(x_i^{(t)})}^{(t)})$ is a local loss, $\Omega(P) = \frac{1}{T(T-1)}\sum_{t=1}^{T}\sum_{s=1,s\neq t}^{T} d(P^{(t)}, P^{(s)})$ is a task regularization incorporating relationship among tasks. And $\lambda \geq 0$ is a free parameter, which allows the loss function to be balanced between the local loss functions and the task regularization. $d(P^{(t)}, P^{(s)})$ reflects the relationship between clusters of task $t$ and $s$, it is deduced from earth mover distance (EMD) (Rubner, Tomasi, and Guibas 1998) and defined as

$$\begin{aligned} d(P^{(t)}, P^{(s)}) &= \min_{W^{ts}} \sum_{z=1}^{c^{(t)}}\sum_{l=1}^{c^{(s)}} w_{zl}^{ts} d_\phi(m_z^{(t)}||m_l^{(s)}) \\ s.t. \quad \sum_{z=1}^{c^{(t)}} w_{zl}^{ts} &= \pi_l^s, \sum_{l=1}^{c^{(s)}} w_{zl}^{ts} = \pi_z^t, w_{zl}^{ts} \geq 0, \forall z, l \end{aligned} \tag{2}$$

where $W^{ts}$ is a nonnegative matrix of size $c^{(t)} \times c^{(s)}$, and $w_{zl}^{ts}$ is the correlation coefficient between $m_z^{(t)}$ and $m_l^{(s)}$. In the constraints, $\pi_z^t = \frac{n_z^{(t)}}{n^{(t)}}$ is the proportion of cluster $z$ in the data corpus $X^{(t)}$, and $\pi_l^s = \frac{n_l^{(s)}}{n^{(s)}}$ is the proportion of cluster $l$ in $X^{(s)}$. $W^{ts}$ can be considered as a joint probability matrix between clusters of task $t$ and task $s$.

### Negative Effect of MBC

The second term in Eq.(1) minimizes the difference between the partitions of any two tasks, it is designed to boost the clustering. The boosting may have positive effect, i.e., helps the current cluster centroid approach the optimal one. It may also have negative effect, i.e., causes the current cluster centroid to deviate from the optimal one. We analyze the effect as follows.

The difference is measured by a distribution metric EMD, which means minimizing the difference between the partitions of any two tasks is equivalent to minimizing the difference between the distributions. Assume the Bregman divergence we use is squared Euclidean distance, (Zhang and Zhang 2010) get the optimal $m_z^{(t)}$ by minimizing Eq.(1)

$$m_z^{(t)} = \frac{A \cdot u_L + B \cdot u_R}{A + B} \tag{3}$$

where $A = \frac{n_z^{(t)}}{n^{(t)}} + \frac{\lambda}{c^{(t)}}$, $B = \frac{\lambda}{c^{(t)}}$, and

$$u_L = \frac{1}{A}\left(\frac{1}{n^{(t)}}\sum_{i\in I_z^{(t)}} x_i^{(t)} + \frac{\lambda}{T-1}\sum_{s\neq t}^{T}\sum_{l=1}^{c^{(s)}} w_{zl}^{ts} m_l^{(s)}\right),$$

$$u_R = (\nabla\phi)^{-1}\left(\frac{\lambda}{B\cdot(T-1)}\sum_{s\neq t}^{T}\sum_{l=1}^{c^{(s)}} w_{zl}^{ts}\nabla\phi(m_l^{(s)})\right).$$

If we set $\lambda = 0$, we can get $m_z^{(t)} = \frac{1}{n_z^{(t)}}\sum_{i\in I_z^{(t)}} x_i^{(t)}$, which is just the centroid calculated by K-means. In Eq.(3), MBC utilizes the relationship $w_{zl}^{ts}$ between the clusters of task $t$ and task $s$ to affect the centroid $m_z^{(t)}$ calculated by K-means. From the analysis above, minimizing the second term has positive effect on the data points shared by other tasks to get their optimal centroids, and negative effect on ones that are not shared by other tasks.

For example, suppose we have two tasks: A and B, the data points of the two tasks are in a two-dimensional space (see Figure 1). Task A is composed of three clusters: a, b and c, task B is composed of three clusters: a, b and d, i.e., the two tasks share the data points in cluster a and b. Assume the two tasks have been clustered correctly, the optimal centroids are $m_a$, $m_b$, $m_c$ and $m_d$ (the solid points in Figure 1). As the second term of MBC need minimize the difference among the partitions of all the tasks, we can get the optimal $W$ by Eq.(2) with standard linear programming techniques to affect the centroids, the computed $W$ is expressed as

$$W = \begin{pmatrix} 0.2133 & 0.0000 & 0.1200 \\ 0.0000 & 0.2133 & 0.1200 \\ 0.1200 & 0.1200 & 0.0933 \end{pmatrix} \tag{4}$$

The centroids of task A and task B are recomputed according to $W$ by Eq.(3) (the hollow points in Figure 1). Obviously, the $W$ in Eq.(4) is not a real optimal result. For example, the weight between the centroids of cluster c in task A and cluster d in task B is 0.0933. As cluster c in task A and cluster d in task B have no relationship, the weight between them is supposed to be zero. However, they will influence each other because of the constraints in Eq.(2), which will bring a negative effect to the optimal centroids selection.

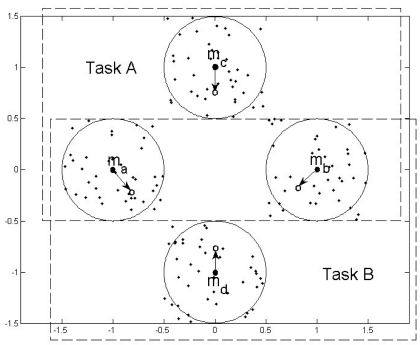


Figure 1: The recomputed centroids of task A and task B, the arrow means the centroid skewing direction

## Smart Multi-task Bregman Clustering

### Avoiding Negative Effect

We compare the local loss of each task calculated by MBC with one calculated by single-task Bregman divergence clustering to judge whether the negative effect occurs. More specifically, in each iteration, we firstly compute the corresponding centroids corpora $M^{(t)}$ and assigning function $h^{(t)}$ through single-task Bregman clustering algorithm and the MBC algorithm considered the task regularization respectively for each task $t$. Secondly we calculate the local loss function in Eq.(1) with $M^{(t)}$ and $h^{(t)}$ of both methods. Thirdly we choose the $M^{(t)}$ and $h^{(t)}$ whose corresponding method has a smaller local loss. Then we recompute the $M^{(t)}$ and $h^{(t)}$ iteratively until $J_{st} = \frac{1}{T}\sum_{t=1}^{T} L^{(t)}(P^{(t)}, X^{(t)})$ converges to a stable state. In this way, we can prevent centroids skewing caused by MBC and get the optimal centroids.

### Optimization

Due to space limitation, we simply introduce the optimization problem of the S-MBC algorithm. The optimization of MBC is a part of the S-MBC optimization problem.

**Computation of** $W$: Given $M$ and $H$, each matrix $W^{ts}$ is independently determined by Eq.(2). This problem can be easily solved by standard linear programming techniques.

**Computation of** $M$:

1) Given $W$ and $H$, we consider the situation when the clustering centroids $M = \{M^{(t)}\}_{t=1}^{T}$ are resulted by the task regularization first. The optimal $m_z^{(t)}$ is computed by

$$\min_{m_z^{(t)}} A \cdot d_\phi(u_L||m_z^{(t)}) + B \cdot d_\phi(m_z^{(t)}||u_R) \qquad (5)$$

where $A$, $B$, $u_L$ and $u_R$ is defined in Eq.(3).

2) When the clustering centroids $M$ is obtained without considering the task regularization, optimizing $m_z^{(t)}$ is equivalent to minimize Eq.(5) with $\lambda = 0$. The optimal $m_z^{(t)}$ is calculated by

$$\min_{m_z^{(t)}} A \cdot d_\phi(u_L||m_z^{(t)}) \qquad (6)$$

where $u_L = \frac{1}{n_z^{(t)}} \sum_{i \in I_z^{(t)}} x_i^{(t)}$.

**Computation of** $H$: Given $M$ and $W$, each assigning function $h^{(t)}$ for task $t$ is independently determined by

$$h^{(t)}(x_i^{(t)}) = \arg\min_z \sum_{i=1}^{n^{(t)}} d_\phi(x_i^{(t)}||m_z^{(t)}). \qquad (7)$$

The overall process is listed in Algorithm 1.

**Algorithm 1:** Smart Multi-task Bregman Clustering(S-MBC)

**Input**: $T$ tasks, $\{X^{(t)}\}_{t=1}^{T}$, clustering numbers of all tasks $\{c^{(t)}\}_{t=1}^{T}$, parameter $\lambda \geq 0$.
**Output**: Clustering assignments $H$.
**Initialization:** Initialize clustering assignments $H$ of all tasks, and compute $M$ according to $H$ without considering task regularization.

**repeat**
  Update $W$ by solving the linear programming of Eq.(2).
  **for** $t = 1$ to $T$ **do**
    Update each $m_z^{(t)}$ of $M^{(t)}$ by Eq.(5).
    Update $h^{(t)}$ with $M^{(t)}$ in step 4 by Eq.(7).
    Calculate $L^{(t)}(P^{(t)}, X^{(t)})$ in Eq.(1) as $J_1$ according to $M^{(t)}$ in step 4 and $h^{(t)}$ in step 5.
    Update each $m_z^{(t)}$ of $M^{(t)}$ by Eq.(6).
    Update $h^{(t)}$ with $M^{(t)}$ in step 7 by Eq.(7).
    Calculate $L^{(t)}(P^{(t)}, X^{(t)})$ in Eq.(1) as $J_2$ according to $M^{(t)}$ in step 7 and $h^{(t)}$ in step 8.
    **if** $J_1 \leq J_2$ **then**
      Save $M^{(t)}$ in step 4 and $h^{(t)}$ computed in step 5.
    **else**
      Save $M^{(t)}$ in step 7 and $h^{(t)}$ computed in step 8.
    **end if**
  **end for**
**until** $J_{st} = \frac{1}{T}\sum_{t=1}^{T} L^{(t)}(P^{(t)}, X^{(t)})$ is convergent.

## Smart Multi-task Kernel Clustering

In this section, we introduce a smart multi-task kernel clustering (S-MKC) algorithm. It can apply to any kind of Mercer kernel (Saunders et al. 1998).

### Problem Formulation

Suppose we are given $T$ clustering tasks, each with a set of points, i.e., $X^{(t)} = \{x_1^{(t)}, x_2^{(t)}, ...., x_{n^{(t)}}^{(t)}\} \in R^d$, $1 \leq t \leq T$, where $n^{(t)}$ is the number of data points in the $t$-th task, and $X = \{X^{(1)}, ..., X^{(T)}\}$ denotes all data corpora. Each data corpus is to be partitioned into $c^{(t)}$ clusters. For each task $t$, we need to find a partition $P^{(t)} = \{M^{(t)}, Z^{(t)}\}$, which is defined by a set of centroids $M^{(t)} = \{m_1^{(t)}, ..., m_{c^{(t)}}^{(t)}\} \in R^{d \times c^{(t)}}$ and a partition matrix $Z^{(t)} \in \{0,1\}^{n^{(t)} \times c^{(t)}}$. $P = \{P^{(t)}\}_{t=1}^{T}$ denotes all the partitions, $M = \{M^{(t)}\}_{t=1}^{T}$ denotes the set of all the centroids, and $Z = \{Z^{(t)}\}_{t=1}^{T}$ denotes all the partition matrixes. We consider a nonlinear mapping $\phi$ from data space to feature space $F$ such that $\phi : R^d \rightarrow$

$F$, the inner product in $F$ is defined as $\langle \phi(x), \phi(y) \rangle_F = K(x, y)$. Then for each task $t$, $X^{(t)}$ mapped in $F$ is defined as $\phi(\mathrm{X^{(t)}}) = \{\phi(x_1^{(t)}), ..., \phi(x_{n^{(t)}}^{(t)})\}$, $1 \leq t \leq T$.

## Objective

We use the multi-task framework in Eq.(1), but inherit the local loss function for Kernel K-means in the matrix form

$$L^{(t)}(P^{(t)}, X^{(t)}) = \frac{1}{n^{(t)}} \left\| \phi(X^{(t)}) - M^{(t)} Z^{(t)T} \right\|_F^2 \quad s.t. \quad Z^{(t)} \in \{0,1\}^{n^{(t)} \times c^{(t)}} \tag{8}$$

where $\|\cdot\|_F$ is a Frobenius norm. And the task regularization is defined as

$$\Omega(P) = \frac{1}{T(T-1)} \sum_{t=1}^{T} \sum_{s=1, s \neq t}^{T} d(P^{(t)}, P^{(s)}) \tag{9}$$

where $d(P^{(t)}, P^{(s)})$ is defined in Eq.(2).

## Optimization

In Eq.(8), the elements in $Z^{(t)}$ can only take binary values, which makes the minimization in Eq.(1) very difficult, (Gordon and Henderson 1977) has given a complete proof that $Z^{(t)}$ can be relaxed into nonnegative continuous domain. We relax $\pi_z^t = \frac{1}{c^{(t)}}$ and $\pi_l^s = \frac{1}{c^{(s)}}$ (Zhang and Zhang 2010). Then according to Eq.(8-9),we obtain the following objective function

$$\begin{aligned} \min_{P,W} J = & \sum_{t=1}^{T} \frac{1}{n^{(t)}} \left\| \phi(X^{(t)}) - M^{(t)} Z^{(t)T} \right\|_F^2 \\ & + \frac{2\lambda}{(T-1)} \sum_{t \neq s}^{T} \sum_{z=1}^{c^{(t)}} \sum_{l=1}^{c^{(s)}} w_{zl}^{ts} ||m_z^{(t)} - m_l^{(s)}||_2^2 \\ & s.t. Z^{(t)} \geq 0, \sum_{z=1}^{c^{(t)}} w_{zl}^{ts} = \frac{1}{c^{(s)}}, \sum_{l=1}^{c^{(s)}} w_{zl}^{ts} = \frac{1}{c^{(t)}}, w_{zl}^{ts} \geq 0. \end{aligned} \tag{10}$$

Minimizing Eq.(10) is with respect to three groups of variables, i.e., cluster centroids $M$, partition matrices $Z$, and relation matrices $W = \{W^{ts}\}$.

**Computation of** $M$: 1) Given $Z$ and $W$, optimizing Eq.(10) with respect to $M$ is equivalent to optimizing

$$\begin{aligned} \min & \sum_{t=1}^{T} \frac{1}{n^{(t)}} \left\| \phi(X^{(t)}) - M^{(t)} Z^{(t)T} \right\|_F^2 \\ & + \frac{2\lambda}{(T-1)} \sum_{t \neq s}^{T} (tr(E^{(t)} (M^{(t)})^T (M^{(t)})) \\ & + tr(E^{(s)} (M^{(s)})^T (M^{(s)})) - 2tr(W^{ts} (M^{(s)})^T (M^{(t)}))) \end{aligned} \tag{11}$$

where $E^{(t)}$ is a diagonal matrix with $E_{ii}^{(t)} = \frac{1}{c^{(t)}} (i = 1, ..., c^{(t)})$. Fixing $\{M^{(s)}\}_{s \neq t}^T$, optimizing Eq.(11) with respect to $M^{(t)}$ is equivalent to optimizing

$$\begin{aligned} J_1 = & \frac{1}{n^{(t)}} \left\| \phi(X^{(t)}) - M^{(t)} Z^{(t)T} \right\|_F^2 \\ & + \frac{2\lambda}{T-1} (tr(E^{(t)} (M^{(t)})^T M^{(t)}) \\ & + \sum_{s \neq t}^{T} (tr(E^{(s)} (M^{(s)})^T M^{(s)}) - 2tr(W^{ts} (M^{(t)})^T M^{(s)})). \end{aligned} \tag{12}$$

Setting $\frac{\partial J_1}{\partial M^{(t)}} = 0$, we obtain

$$\begin{aligned} M^{(t)} = & (\frac{1}{n^{(t)}} \phi(X^{(t)}) Z^{(t)} + \frac{2\lambda}{T-1} \sum_{s \neq t}^{T} M^{(s)} W^{st}) \\ & (\frac{1}{n^{(t)}} Z^{(t)T} Z^{(t)} + \frac{2\lambda}{T-1} E^{(t)})^{-1}. \end{aligned} \tag{13}$$

In Eq.(13), $\phi(X^{(t)})$ and $M^{(s)}$ are unknown, however, $\phi(X^{(s)})^T M^{(t)}$ and $(M^{(s)})^T M^{(t)}$ can be calculated. Setting $F^{(s,t)} = \phi(X^{(s)})^T M^{(t)}$, $G^{(s,t)} = (M^{(s)})^T M^{(t)} (t, s = 1, ..., T)$, then optimizing $M^{(t)}$ is equivalent to optimizing $F^{(s,t)}$ and $G^{(s,t)} (s = 1, ..., T)$. Setting $K^{(s,t)} = \phi(X^{(s)})^T \phi(X^{(t)})$, we obtain

$$\begin{aligned} F^{(s,t)new} = & (\frac{1}{n^{(t)}} K^{(s,t)} Z^{(t)} + \frac{2\lambda}{T-1} \sum_{p \neq t}^{T} F^{(s,p)} W^{pt}) \\ & (\frac{1}{n^{(t)}} Z^{(t)T} Z^{(t)} + \frac{2\lambda}{T-1} E^{(t)})^{-1}, \end{aligned} \tag{14}$$

$$\begin{aligned} G^{(s,t)new} = & (\frac{1}{n^{(s)}} Z^{(s)T} Z^{(s)} + \frac{2\lambda}{T-1} E^{(s)})^{-1} V \\ & (\frac{1}{n^{(t)}} Z^{(t)T} Z^{(t)} + \frac{2\lambda}{T-1} E^{(t)})^{-1} \end{aligned} \tag{15}$$

where $V = \frac{1}{n^{(s)} n^{(t)}} Z^{(s)T} K^{(s,t)} Z^{(t)}$

$$\begin{aligned} & + \frac{2\lambda}{n^{(s)}(T-1)} \sum_{p \neq t}^{T} Z^{(s)T} F^{(s,p)} W^{pt} \\ & + \frac{2\lambda}{n^{(t)}(T-1)} \sum_{q \neq s}^{T} W^{sq} F^{(t,q)T} Z^{(t)} \\ & + \frac{4\lambda^2}{(T-1)^2} \sum_{q \neq s}^{T} \sum_{p \neq t}^{T} W^{sq} G^{(q,p)} W^{pt}. \end{aligned}$$

2) When the clustering centroids $M$ is obtained without considering the task regularization, optimizing $M^{(t)}$ is equivalent to minimize Eq.(14) and Eq.(15) with $\lambda = 0$.

**Computation of** $Z$: Given $M$ and $W$, each $Z^{(t)}$ can be obtained by optimizing

$$\min \left\| \phi(X^{(t)}) - M^{(t)} Z^{(t)T} \right\|_F^2 \quad s.t. \quad Z^{(t)} \geq 0. \tag{16}$$

For the constraint $Z^{(t)} \geq 0$, we cannot get a closed form solution of $Z^{(t)}$. In the following, we introduce the

Lagrangian multiplier $\gamma \in R^{n^{(t)} \times c^{(t)}}$, and the Lagrangian function is

$$L(Z^{(t)}) = \left\| \phi(X^{(t)}) - M^{(t)} Z^{(t)T} \right\|_F^2 - tr(\gamma Z^{(t)T}). \quad (17)$$

Setting $\frac{\partial L(Z^{(t)})}{\partial Z^{(t)}} = 0$, we obtain $\gamma = -2A + 2Z^{(t)}B$, where $A = \phi(X^{(t)})^T M^{(t)\phi} = F^{(t,t)new}$, $B = (M^{(t)\phi})^T M^{(t)\phi} = G^{(t,t)new}$.

Using the Karush-Kuhn-Tucker condition $\gamma_{ij} Z_{ij}^{(t)} = 0$ (Boyd and Vandenberghe 2004), we get $[-A + Z^{(t)}B]_{ij} Z_{ij}^{(t)} = 0$. Introduce $A = A^+ - A^-$ and $B = B^+ - B^-$, where $A_{ij}^+ = (|A_{ij}| + A_{ij})/2$ and $A_{ij}^- = (|A_{ij}| - A_{ij})/2$ (Ding, Li, and Jordan 2008), we obtain

$$[A^- + Z^{(t)}B^+ - A^+ - Z^{(t)}B^-]_{ij} Z_{ij}^{(t)} = 0. \quad (18)$$

Eq.(18) leads to the following updating formula

$$Z_{ij}^{(t)} \leftarrow Z_{ij}^{(t)} \sqrt{\frac{[A^+ + Z^{(t)}B^-]_{ij}}{[A^- + Z^{(t)}B^+]_{ij}}} \quad (19)$$

**Computation of** $W$: Given $M$ and $Z$, each matrix $W^{ts}$ is independently determined by

$$\begin{aligned} &\min_{W^{ts}} \sum_{z=1}^{c^{(t)}} \sum_{l=1}^{c^{(s)}} w_{zl}^{ts} ||m_z^{(t)} - m_l^{(s)}||_2^2 \\ s.t. \quad &\sum_{z=1}^{c^{(t)}} w_{zl}^{ts} = \pi_l^s, \sum_{l=1}^{c^{(s)}} w_{zl}^{ts} = \pi_z^t, w_{zl}^{ts} \geq 0, \forall z, l. \end{aligned} \quad (20)$$

This problem can be efficiently solved by standard linear programming techniques. $||m_z^{(t)} - m_l^{(s)}||_2^2$ in Eq.(20) can be calculated by

$$\begin{aligned} &||m_z^{(t)} - m_l^{(s)}||_2^2 \\ &= (m_z^{(t)})^T (m_z^{(t)}) - 2(m_z^{(t)})^T (m_l^{(s)}) + (m_l^{(s)})^T (m_l^{(s)}) \\ &= G^{(t,t)}(z,z) - 2G^{(t,s)}(z,l) + G^{(s,s)}(l,l). \end{aligned} \quad (21)$$

The local loss function in Eq.(8) can be calculated by

$$\begin{aligned} L^{(t)}(P^{(t)}, X^{(t)}) = \frac{1}{n^{(t)}} tr(&K^{(t,t)} - F^{(t,t)} Z^{(t)T} \\ &- Z^{(t)} F^{(t,t)T} + Z^{(t)} G^{(t,t)} Z^{(t)T}). \end{aligned} \quad (22)$$

The overall process of S-MKC is listed in Algorithm 2.

## Experiments

We compare the proposed multi-task clustering methods S-MBC and S-MKC, with typical single-task clustering methods: K-means (KM) and Kernel K-means (KKM), and typical multi-task clustering method: MBC. To evaluate the clustering results, we adopt two performance measures in (Xu, Liu, and Gong 2003): clustering accuracy (Acc) and normalized mutual information (NMI), which are widely used in the literature.

**Algorithm 2:** Smart Multi-task Kernel Clustering (S-MKC)

**Input**: $T$ tasks, $\{X^{(t)}\}_{t=1}^T$, clustering numbers of all tasks $\{c^{(t)}\}_{t=1}^T$, parameter $\lambda \geq 0$.
**Output**: Clustering assignments $\{Z^{(t)}\}_{t=1}^T$.
**Initialization:** Initialize clustering assignments $Z$ of all tasks, and compute $M$ according to $Z$ without considering task regularization. Compute the kernel matrix K using a specific Mercer kernel.

```
repeat
    Update W by solving the linear programming of Eq.(20).
    for t = 1 to T do
        Update F^(s,t) and G^(s,t) by Eq.(14) and Eq.(15).
        Update Z^(t) with F^(t,t) and G^(t,t) in step 4 by Eq.(19).
        Calculate L^(t)(P^(t), X^(t)) in Eq.(22) as O_1 according
        to F^(t,t) and G^(t,t) in step 4 and Z^(t) in step 5.
        Update F^(s,t) and G^(s,t) by Eq.(14) and Eq.(15)
        respectively with λ = 0.
        Update Z^(t) with F^(t,t) and G^(t,t) in step 7 by Eq.(19).
        Calculate L^(t)(P^(t), X^(t)) in Eq.(22) as O_2 according
        to F^(t,t) and G^(t,t) in step 7 and Z^(t) in step 8.
        if O_1 ≤ O_2 then
            Save F^(s,t) and G^(s,t) in step 4 and Z^(t) in step 5.
        else
            Save F^(s,t) and G^(s,t) in step 7 and Z^(t) in step 8.
        end if
    end for
until J_st = (1/T) Σ_{t=1}^T L^(t)(P^(t), X^(t)) is convergent.
```

Table 1: Data Sets

| Data set | Task id | #Sample | #Feature | Class |
|---|---|---|---|---|
| NG1 | Task 1 | 1493 | 43586 | 3(6-8) |
| | Task 2 | 1494 | 43586 | 3(6-8) |
| NG2 | Task 1 | 1600 | 43586 | 4(3, 4, 12, 15) |
| | Task 2 | 1600 | 43586 | 2 |
| NG3 | Task 1 | 4000 | 43586 | 10(1-10) |
| | Task 2 | 4800 | 43586 | 12(4-15) |
| | Task 3 | 6400 | 43586 | 16(5-20) |
| Reviews | Task 1 | 3520 | 18482 | 3(1-3) |
| | Task 2 | 2658 | 18482 | 3(2-4) |
| | Task 3 | 1937 | 18482 | 3(3-5) |

## Data Sets

We use 2 data sets 20Newsgroups and Reviews in (Zhong and Ghosh 2003)[1].

Original 20Newsgroups[2] data set is composed of 6 root categories, under which are 20 sub categories. We use three splitting schemes to construct 3 data sets to demonstrate three typical cases of multi-task clustering.

1) The first case is that all data sets are from a same distribution. We construct NG1 to represent this case, by splitting

[1] http://www.shi-zhong.com/software/docdata.zip.
[2] http://kdd.ics.uci.edu/databases/20newsgroups.

Table 2: Clustering Results on NG1

| Method | Task id | Acc(%) | NMI(%) |
|---|---|---|---|
| KM | Task 1 | $34.8091 \pm 2.7814$ | $14.1337 \pm 3.3576$ |
| | Task 2 | $33.7282 \pm 0.1450$ | $12.5061 \pm 0.5498$ |
| MBC | Task 1 | $52.3538 \pm 6.9629$ | $26.7529 \pm 0.6507$ |
| | Task 2 | $49.7456 \pm 4.5002$ | $27.2322 \pm 5.5268$ |
| S-MBC | Task 1 | $52.7482 \pm 5.6311$ | $27.6567 \pm 0.2784$ |
| | Task 2 | $50.6158 \pm 5.9836$ | $28.7772 \pm 5.0728$ |
| KKM | Task 1 | $72.9190 \pm 3.2077$ | $42.1187 \pm 3.8203$ |
| | Task 2 | $74.8126 \pm 5.8943$ | $44.9751 \pm 8.5358$ |
| S-MKC | Task 1 | $\mathbf{79.0308 \pm 2.6404}$ | $\mathbf{45.4786 \pm 3.0120}$ |
| | Task 2 | $\mathbf{80.3186 \pm 5.8329}$ | $\mathbf{49.0151 \pm 6.1227}$ |

Table 3: Clustering Results on NG2

| Method | Task id | Acc(%) | NMI(%) |
|---|---|---|---|
| KM | Task 1 | $25.1085 \pm 4.9994$ | $10.1312 \pm 7.0321$ |
| | Task 2 | $49.6680 \pm 0.1079$ | $18.2581 \pm 0.4278$ |
| MBC | Task 1 | $42.3375 \pm 3.2312$ | $20.4846 \pm 7.8106$ |
| | Task 2 | $54.0250 \pm 2.7075$ | $23.6679 \pm 1.1445$ |
| S-MBC | Task 1 | $44.8375 \pm 3.2854$ | $21.1936 \pm 4.8404$ |
| | Task 2 | $54.3500 \pm 1.0380$ | $24.0475 \pm 1.3159$ |
| KKM | Task 1 | $60.0875 \pm 6.7322$ | $27.2518 \pm 3.5686$ |
| | Task 2 | $78.8750 \pm 4.3853$ | $28.4355 \pm 7.3585$ |
| S-MKC | Task 1 | $\mathbf{71.8333 \pm 4.2864}$ | $\mathbf{40.8279 \pm 6.6604}$ |
| | Task 2 | $\mathbf{84.6257 \pm 2.5977}$ | $\mathbf{37.7968 \pm 7.4034}$ |

the selected documents into 2 parts, with each part having 3 sub categories.

2) The second case is that all tasks are on an identical data set but requires clusters at different resolutions. NG2 is constructed to represent this case. We randomly select 400 documents in class 3,4,12,15, and combine the selected documents as the identical data set, which also belong to Comp and Sci root categories.

3) The third case is that the distributions of all tasks are not identical but similar, the tasks share some data sets from the same class labels. We construct NG3 and Reviews to represent the case. In NG3, we randomly select 400 documents in each sub category.

## Parameter Settings

For MBC, S-MBC and S-MKC, we set the same $\lambda = 0.5$ for all the algorithms to ensure a fair comparison for them and make all the algorithms reach a good performance. The Bregman divergence we choose is Euclidean distance. We use the Gaussian kernel function for S-MKC to compute the kernel matrix as its good separability, the width of the Gaussian kernel $\sigma$ is set by searching the grid $\{0.1, 1, 10, 100\}$. Under each parameter setting, we repeat clustering 10 times, and the mean result as well as the standard deviation is computed. We report the mean and standard deviation corresponding to the best parameter setting for each method to compare with each other.

## Clustering Results

The average results for all the data sets in Table 1 are shown in Table 2, Table 3, Table 4 and Table 5. We can summarize the following points from the clustering results.

1) When the tasks are from a same distribution (Table 2 and Table 3), MBC can get a fairly good clustering result compared with single-task clustering algorithm K-means. S-MBC improves a little upon MBC.

2) When the distributions of all tasks are not identical but similar (Table 4 and Table 5), MBC cannot get a very good clustering result, since it has negative effect on the cluster-updating process. While S-MBC improves much upon MBC by avoiding the negative effect.

Table 4: Clustering Results on NG3

| Method | Task id | Acc(%) | NMI(%) |
|---|---|---|---|
| KM | Task 1 | $11.8700 \pm 0.7079$ | $6.0105 \pm 1.3500$ |
| | Task 2 | $11.9375 \pm 1.1324$ | $7.9813 \pm 1.0814$ |
| | Task 3 | $10.0875 \pm 1.1639$ | $9.8231 \pm 1.8123$ |
| MBC | Task 1 | $15.0500 \pm 0.2577$ | $6.6964 \pm 1.3796$ |
| | Task 2 | $15.2917 \pm 0.5387$ | $7.8704 \pm 2.1348$ |
| | Task 3 | $22.7813 \pm 0.4361$ | $18.6318 \pm 1.4769$ |
| S-MBC | Task 1 | $23.6450 \pm 1.0470$ | $16.2028 \pm 2.2518$ |
| | Task 2 | $30.2500 \pm 1.8372$ | $25.3250 \pm 2.0037$ |
| | Task 3 | $25.5938 \pm 1.5998$ | $21.4389 \pm 1.8458$ |
| KKM | Task 1 | $34.8500 \pm 1.8365$ | $24.6034 \pm 1.4740$ |
| | Task 2 | $34.2917 \pm 0.9169$ | $24.4588 \pm 2.6256$ |
| | Task 3 | $30.8125 \pm 1.8455$ | $26.0923 \pm 1.9551$ |
| S-MKC | Task 1 | $\mathbf{41.7500 \pm 1.7170}$ | $\mathbf{27.5468 \pm 2.6612}$ |
| | Task 2 | $\mathbf{42.2917 \pm 2.2997}$ | $\mathbf{28.4778 \pm 1.8052}$ |
| | Task 3 | $\mathbf{38.8750 \pm 1.1862}$ | $\mathbf{27.2567 \pm 1.2729}$ |

3) For nonlinear separable data sets such as NG1, NG2, NG3, both single-task Kernel clustering (KKM) and multi-task Kernel clustering (S-MKC) perform much better. S-MKC improves upon KKM since it takes advantages of positive effect of cluster-relationship boosting and avoids negative effect at the same time. However, for the linear separable data Reviews, S-MBC performs the best.

## Conclusion

In this paper, we have proposed two improved algorithms S-MBC and S-MKC based on MBC. S-MBC uses a task loss to identify the negative effect brought by MBC, and avoids the negative effect when it occurs. S-MKC can deal with nonlinear separable data by a specific optimization method, and be applied to any Mercer kernel. Experimental results confirm our analysis, and demonstrate the superiority of our proposed methods.

Table 5: Clustering Results on Reviews

| Method | Task id | Acc(%) | NMI(%) |
|---|---|---|---|
| KM | Task 1 | $46.6761 \pm 5.6301$ | $26.7293 \pm 6.6463$ |
| | Task 2 | $58.0625 \pm 5.9307$ | $27.5920 \pm 5.7872$ |
| | Task 3 | $62.0960 \pm 9.5752$ | $19.9361 \pm 7.3979$ |
| MBC | Task 1 | $52.5653 \pm 5.6894$ | $23.9243 \pm 6.1550$ |
| | Task 2 | $50.7336 \pm 8.6716$ | $24.8309 \pm 8.7917$ |
| | Task 3 | $67.2272 \pm 4.0025$ | $20.1917 \pm 5.6629$ |
| S-MBC | Task 1 | $\mathbf{73.2892 \pm 3.1169}$ | $\mathbf{43.3727 \pm 5.1202}$ |
| | Task 2 | $\mathbf{79.7239 \pm 3.9224}$ | $\mathbf{46.9371 \pm 5.9629}$ |
| | Task 3 | $\mathbf{78.8508 \pm 5.5063}$ | $\mathbf{37.0873 \pm 7.4804}$ |
| KKM | Task 1 | $51.1307 \pm 6.6646$ | $23.8308 \pm 6.2104$ |
| | Task 2 | $60.8751 \pm 9.0664$ | $27.9711 \pm 8.9459$ |
| | Task 3 | $63.5674 \pm 8.2900$ | $32.2453 \pm 9.9521$ |
| S-MKC | Task 1 | $73.1631 \pm 3.9026$ | $40.5145 \pm 3.9577$ |
| | Task 2 | $78.0986 \pm 6.6804$ | $42.6631 \pm 5.0044$ |
| | Task 3 | $72.9824 \pm 2.5606$ | $37.0017 \pm 3.7449$ |

## Acknowledgments

This work was supported by National Science Foundation of China (No. 61272374), Program for New Century Excellent Talents in University (NCET) of China (No. NCET-11-0056), Specialized Research Fund for the Doctoral Program of Higher Education (No.20120041110046) and Key Project of Chinese Ministry of Education(No. 313011).

## References

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41-75.

Ando, R. K., and Zhang, T. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* 6:1817-1853.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2006. Multi-task feature learning. In *NIPS*, 41-48.

Micchelli, C. A., and Pontil, M. 2004. Kernels for multi-task learning. In *NIPS*.

Micchelli, C. A., and Pontil, M. 2005. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research* 6:615-637.

Gu, Q., and Zhou, J. 2009. Learning the shared subspace for multi-task clustering and transductive transfer classification. In *ICDM*, 159-168.

Gu, Q.; Li, Z.; and Han, J. 2011. Learning a kernel for multi-task clustering. In *AAAI*, 368-373.

Zhang, J., and Zhang, C. 2010. Multitask Bregman clustering. In *AAAI*, 655-660.

Girolami, M. 2002. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks* 13:780-784.

Rubner, Y.; Tomasi, C.; and Guibas, L. 1998. A metric for distributions with applications to image databases. In *ICCV*, 59-66.

Nielsen, F., and Nock, R. 2009. Sided and symmetrized Bregman centroids. *IEEE Transactions on Information Thoery* 55(6):2882-2904.

Saunders, C.; Stitson, M. O.; Weston, J.; Bottou, L.; Schö lkopf, B.; and Smola, A. 1998. Support vector machine reference manual. Technical Report, CSD-TR-98-03, Dept. of Computer Science, Royal Holloway College.

Gordon, A. D., and Henderson, J. T. 1977. An algorithm for Euclidean sum-of-squares classification. *Biometrics* 33:355-362.

Boyd, S., and Vandenberghe, L. 2004. Convex optimization. Cambridge, UK: Cambridge University Press.

Ding, C. H.; Li, T.; and Jordan, M. I. 2008. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99(1):45-55.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *SIGIR*, 267-273.

Zhong, S., and Ghosh, J. 2003. A unified framework for model for model-based clustering. *Journal of Machine Learning Research* 4:1001-1037.