Multitask Bregman Clustering[*]

Jianwen Zhang and **Changshui Zhang**
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Automation, Tsinghua University, Beijing 100084, China
jw-zhang06@mails.tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn

Abstract

Traditional clustering methods deal with a single clustering task on a single data set. However, in some newly emerging applications, multiple similar clustering tasks are involved simultaneously. In this case, we not only desire a partition for each task, but also want to discover the relationship among clusters of different tasks. It's also expected that the learnt relationship among tasks can improve performance of each single task. In this paper, we propose a general framework for this problem and further suggest a specific approach. In our approach, we alternatively update clusters and learn relationship between clusters of different tasks, and the two phases boost each other. Our approach is based on the general Bregman divergence, hence it's suitable for a large family of assumptions on data distributions and divergences. Empirical results on several benchmark data sets validate the approach.

Introduction

Clustering is a fundamental problem in machine learning and data mining. Several famous algorithms have been proposed and successfully implemented, such as *k-means* (Jain and Dubes 1988), spectral clustering (Ng, Jordan, and Weiss 2002), Bregman divergence based clustering (Banerjee et al. 2005), etc.. Traditional clustering methods deal with a single clustering task on a single data set. i.e., an algorithm is required to provide a partition for a given target data set. However, new requirements emerge from more and more complex applications recently. In these applications, we are confronted with multiple similar clustering tasks, which are often on different data sets. Generally, there are three types of typical scenarios. First, we hope to improve individual clustering performance by transferring knowledge across the tasks. The feasibility has been approved by frontier research on *multi-task learning* (Caruana 1997; Ando and Zhang 2005) and *transfer learning* (Pan and Yang 2008). Second, since the tasks are similar, the clustering results on them are expected to be coherent. For example, when clustering at different resolution on a same data set or clustering on slowly time-evolving data (Chakrabarti, Kumar, and Tomkins 2006), significant contradiction certainly leads to perplexity on understanding data. Third, we want to discover the relationship between different data sets via clustering analysis. For example, when analyzing species of different area in ecology or *haplotypes* of different human populations in genetic (Stephens, Smith, and Donnelly 2001), we hope to discover the relationship between clusters of different data sets.

In above scenarios, we have strong prior that there are high correlation among involved data sets. This motivates us to find an approach to perform multiple clustering tasks collaboratively rather than in the traditional isolated manner. The approach is desired to be able to (1) improve the performance of individual clustering tasks; (2) produce results coherent among tasks; (3) discover the relationship between clusters of different tasks. What's more, since there have been lots of widely used clustering algorithms, suitable for different assumption on data distribution or divergence measure, we hope the approach is general to be able to directly extend these traditional algorithms to multitask applications.

Multitask clustering introduced here belongs to the field of *multitask learning* (Caruana 1997; Ando and Zhang 2005; Argyriou et al. 2008). However, compared to standard multitask learning, multitask clustering introduced here has a distinctive feature: it does not only want to improve single task's performance, but also discover the relationship between clusters of different tasks.

In this paper, we propose a general framework for this problem and also suggest a specific approach. In the framework, multitask clustering is formulated as minimizing a loss composed of a task loss and a task regularization. In our approach, each task loss is the average Bregman divergence from a sample to its cluster centroid, and the task regularization is a type of average divergence between partitions of any two tasks. We further propose an alternative method to solve the optimization problem. The method alternatively update clusters and learn relationship between clusters of different tasks, and the two phases boost each other. Empirical results on several benchmark data sets validate our approach.

Related works

Multitask learning (Caruana 1997; Ando and Zhang 2005; Argyriou et al. 2008) aims to perform multiple learning

[*]This work was supported by National Science Foundation of China (NSFC, Grant No. 60835002 and No. 60721003).

Table 1: Frequently used Bregman divergences.

Domain	$\phi(x)$	$d_\phi(x,y)$	Divergence
$\mathbb{R}^d$	$\|x\|^2$	$\|x-y\|^2$	Squared Euclidean distance
$\mathbb{R}^d$	x^TAx	$(x-y)^TA(x-y)$	Mahalanobis distance
d-simplex	$\sum_{j=1}^d x_j \log_2 x_j$	$\sum_{j=1}^d x_j \log\left(\frac{x_i}{y_i}\right)$	KL-divergence
$\mathbb{R}_{++}$	$-\log x$	$\frac{x}{y} - \log\left(\frac{x}{y}\right) - 1$	Itakura-Saito distance

tasks together to improve individual performance. However, almost all existing works focused on supervised settings. (Gu and Zhou 2009) handled the clustering problem by learning a shared subspace among tasks. However, the approach assumes that all tasks share an identical set of clusters and requires that cluster numbers of all tasks are the same, which is too restrictive in practice. Moreover, the approach is especially designed for Euclidean distance, which can not apply to more general case of divergences.

Transfer learning (Pan and Yang 2008) attempts to improve learning performance on a target data set by utilizing auxiliary data sets. Hence in transfer learning, different data sets are not equally treated, which is different from multitask learning. On clustering problems, (Dai et al. 2008) proposed an approach to clustering a small collection of target data with the help of a large amount of unlabeled auxiliary data.

Clustering ensemble (Strehl and Ghosh 2003; Topchy, Jain, and Punch 2005) aims to combine multiple *given* partitions on an *identical* data set to reach a consensus. Rather differently, in multitask clustering, the partitions for all tasks are exactly unknown and desired. Moreover, multiple tasks are often on different data sets.

Preliminaries

Bregman divergence

Clustering problem is often formulated as minimizing certain type of average divergence between a data sample and corresponding cluster centroid. Hence the divergence assumption is crucial to a clustering algorithm. It has been found that a large family of divergences can be written as a uniform form called *Bregman divergences*.

Definition 1 *(Bregman 1967) Let $\mathcal{S} \subset \mathbb{R}^d$ be a convex set with the relative interior $ri(\mathcal{S})$ nonempty, and $\phi : \mathcal{S} \mapsto \mathbb{R}$ be a strictly convex function differentiable on $ri(\mathcal{S})$. The **Bregman divergence** $d_\phi : \mathcal{S} \times ri(\mathcal{S}) \mapsto [0, \infty]$ is defined as*

$$d_\phi(x\|y) = \phi(x) - \phi(y) - \langle x - y, \nabla\phi(y)\rangle ,$$

where $\nabla\phi(y)$ represents the gradient of ϕ evaluated at y.

Tab. 1 lists some frequently used divergences as special cases of Bregman divergences.

A Bregman divergence is not generally symmetric, i.e., it does not always hold that $d_\phi(x\|y) = d_\phi(y\|x)$. Two examples of symmetric Bregman divergences are squared Euclidean distance and Mahalanobis distance. KL divergence is asymmetric. In addition, a Bregman divergence is convex w.r.t. its left variate but non-convex w.r.t. its right variate.

(Banerjee et al. 2005) summarized an uniform formulation for clustering problem with Bregman divergences:

$$\min_{\mathcal{U},h} \frac{1}{n} \sum\nolimits_{i=1}^{n} d_\phi\left(x_i \| u_{h(x_i)}\right), \qquad (1)$$

where $\mathcal{U} = \{u_1, \ldots, u_K\}$ is the set of all cluster centroids, and assigning function $h : \mathcal{X} \mapsto \{1, \ldots, K\}$ maps from a sample x_i to its cluster index. Lots of traditional clustering algorithms such as *k-means*, *Gaussian mixture models*, *model-based clustering* (Zhong and Ghosh 2003) are special cases of this formulation.

Multitask Bregman clustering

Problem and notations

Consider T clustering tasks. Each task t is on a data corpus $\mathcal{X}^t = \{x_1^t, \ldots, x_{n^t}^t\}$, and $\mathcal{X} = \{\mathcal{X}^1, \ldots, \mathcal{X}^T\}$ denotes all data corpora. Each data corpus is to be partitioned into K^t clusters. For each task t, we need to find a partition $\mathcal{P}^t = \{\mathcal{U}^t, h^t\}$, which is defined by an assigning function $h^t : \mathcal{X}^t \mapsto \{1, \ldots, K^t\}$ and a set of centroids $\mathcal{U}^t = \{u_1^t, \ldots, u_{K^t}^t\}$. $\mathcal{P} = \{\mathcal{P}^t\}_{t=1}^T$ denotes all the partitions. $\mathcal{K} = \{K^1, \ldots, K^T\}$ denotes the set of all the clustering numbers. All superscripts ^{t} and ^{s} are task indices rather than power. Subscripts $_z$ and $_l$ indicate cluster indices. Subscript $_i$ indicates index for a data sample.

A general framework

Considering T clustering tasks together, we can formulate the multitask clustering problem as finding a set of partitions $\mathcal{P}$ to minimize following loss function

$$\min_{\mathcal{P}} \mathcal{L} = \frac{1}{T} \sum\nolimits_{t=1}^{T} \mathfrak{L}^t\left(\mathcal{P}^t, \mathcal{X}^t\right) + \lambda\,\Omega\left(\mathcal{P}\right). \qquad (2)$$

In Eq. (2), $\mathfrak{L}^t\left(\mathcal{P}^t, \mathcal{X}^t\right)$ is a local loss for task t, and $\Omega\left(\mathcal{P}\right)$ is a task regularization incorporating relationship among tasks. And $\lambda \geq 0$ is a free parameter.

Eq. (2) is a rather general framework. The choice of local loss is the same as that in traditional clustering problems. The form of task regularization is application dependent, reflecting the properties desired for relationship between tasks in that application problem.

A suggested formulation

Let's revert to the specific problem in this paper. Besides clustering the data of each task well, we also want the results of similar tasks to be coherent and expect that the performance on each single task can be enhanced by utilizing multiple tasks. As to the local loss $\mathfrak{L}^t\left(\mathcal{P}^t, \mathcal{X}^t\right)$, we inherit that for clustering with Bregman divergence, i.e., the average divergence from a data sample to its cluster centroid

$$\mathfrak{L}^t\left(\mathcal{P}^t\right) = \frac{1}{n^t} \sum\nolimits_{i=1}^{n^t} d_\phi\left(x_i^t \| u_{h^t(x_i^t)}^t\right). \qquad (3)$$

As to the task regularization, we suggest following formulation

$$\Omega\left(\mathcal{P}\right) = \frac{1}{T(T-1)} \sum\nolimits_{t=1}^{T} \sum\nolimits_{s=1, s\neq t}^{T} d\left(\mathcal{P}^t, \mathcal{P}^s\right) \qquad (4)$$

to encourage results of different tasks coherent. In Eq. (4), $d\left(\mathcal{P}^t, \mathcal{P}^s\right)$ measures the divergence from the partition $\mathcal{P}^t$ of task t to the partition $\mathcal{P}^s$ of task s . Notice that $d\left(\mathcal{P}^t, \mathcal{P}^s\right)$ is not a Bregman divergence.

A proper form for the divergence between two clustering models is right the difficult problem in multitask clustering. On the one hand, we can not use a metric of functions like $\|h^t - h^s\|$ in classification problems, as indistinguishability about the cluster labels exists for clustering. On the other, we hope the divergence $d\left(\mathcal{P}^t, \mathcal{P}^s\right)$ reflects relationship between clusters of task t and s rather than just a value of distance between two functions. Hence we define the form of $d\left(\mathcal{P}^t, \mathcal{P}^s\right)$ as

$$d\left(\mathcal{P}^t, \mathcal{P}^s\right) = \min_{W^{ts}} \sum\nolimits_{z=1}^{K^t} \sum\nolimits_{l=1}^{K^s} w_{zl}^{ts} d_\phi\left(u_z^t \| u_l^s\right), \tag{5}$$

$$s.t. \quad \sum\nolimits_z w_{zl}^{ts} = \pi_l^s \,, \sum\nolimits_l w_{zl}^{ts} = \pi_z^t \,, w_{zl}^{ts} \geq 0 \,, \forall\, z, l, \tag{6}$$

where W^{ts} is a nonnegative matrix of size $K^t \times K^s$, and w_{zl}^{ts} is the element at row z and column l. This matrix's summations along each row and each column are constrained by Eq. (6). In the constraints, $\pi_z^t = n_z^t / n^t$ is the proportion of cluster z in the data corpus $\mathcal{X}^t$ of task t. Similarly, $\pi_l^s = n_l^s / n^s$ is the proportion of cluster l in $\mathcal{X}^s$ of task s. Obviously, $\sum_z \pi_z^t = \sum_l \pi_l^s = 1$, hence $\sum_{zl} w_{zl}^{ts} = 1$. Thus W^{ts} can be considered as a joint probability matrix between clusters of two tasks, with each marginal equal to the empirical prior on clusters of a task.

The actual meaning of the task regularization based on divergence of Eq. (5) will be further explained in the followed subsection.

Why such a task regularization

In this subsection, we will show that the suggested formulation is equivalent to joint density estimation on multiple data corpora via mixture density models while requiring that learnt density models of different tasks are similar. The task regularization defined by Eq. (5) actually defines a divergence between two learnt density models of tasks t and s. Thus the theoretical foundation of the suggested formulation is made clear.

(Banerjee et al. 2005) showed that there is a bijection between a Bregman divergence d_ϕ and an *exponential family*

$$p_\Psi(x;\theta) = \exp\left\{\langle\theta, x\rangle\rangle - \Psi(\theta)\right\} p_0(x) \tag{7}$$

where θ, x, and $\Psi(\theta)$ are called *natural parameter*, *natural statistic*, and *cumulant function*, respectively. In fact, the density Eq. (7) can be represented using a Bregman divergence as $p_\Psi(x,\theta) = \exp\left\{-d_\phi(x,u)\right\} b_\phi(x)$. The parameters u and θ, functions ϕ and Ψ are linked by $u = \nabla\Psi(\theta)$, $\theta = \nabla\phi(u)$, and $d_\Psi(\theta_1 \| \theta_2) = d_\phi(u_2 \| u_1)$. Moreover, for two distributions p_1 and p_2 in a same exponential family, $\mathrm{KL}(p_1 || p_2) = d_\Psi(\theta_2 \| \theta_1) = d_\phi(u_1 \| u_2)$.

As explained in (Zhang et al. 2009), Bregman clustering via Eq. (1) is equivalent to using an exponential family mixture distribution $g_\Psi(x;\Theta) = \sum_{z=1}^K \alpha_z p_\Psi(x;\theta_z)$ to approximate true data distribution $f(x)$, i.e., $\min_\Theta \mathrm{KL}\left(f \| g_\Psi\right)$.

If we measure divergence between two exponential family mixture distributions, g_Ψ^t and g_Ψ^s, *earth mover distance* (EMD) (Rubner, Tomasi, and Guibas 1998) is a good choice. EMD measures divergence between two discrete distributions, which is widely used in music analysis and computer vision. The EMD between g_Ψ^t and g_Ψ^s is defined as

$$d_M(g_\Psi^t, g_\Psi^s) = \min_W \sum\nolimits_{z=1}^{K^t} \sum\nolimits_{l=1}^{K^s} w_{zl}\, d\left(p_z^t, p_l^s\right), \tag{8}$$

$$s.t. \quad \sum\nolimits_z w_{zl}^{ts} = \alpha_l^s \,, \sum\nolimits_l w_l^{ts} = \alpha_z^t \,, w_{zl}^{ts} \geq 0 \,, \forall\, z, l \,,$$

where we simply use p_z^t to denote $p_\Psi^t(x;\theta_z)$. $d\left(p_z^t, p_l^s\right)$ is a predefined divergence between two component distributions. If we take $d\left(p_z^t, p_l^s\right)$ as a KL divergence, utilizing the property $\mathrm{KL}(p_z^t || p_l^s) = d_\phi(u_z^t \| u_l^s)$, we obtain an interesting result that the EMD defined by Eq. (8) is just the divergence we define in Eq. (5).

Now the meaning of the suggested formulation of Eq. (3-5) is clear: the multitask clustering is equivalent to jointly approximate data distributions of all tasks, each by an exponential family mixture density model. The local loss of Eq. (3) is to require that the local mixture density approximates local data density well. The task regularization defined by Eq. (4-5) is to require the learnt local mixture densities for all tasks are similar to each other.

Solving the optimization problem

In this section, we discuss how to solve the resulted optimization problem.

According to Eq. (2-5), the proposed approach is converted to following optimization problem

$$\min_{\mathcal{P},\mathcal{W}} \mathfrak{L}\left(\mathcal{P},\mathcal{W}\right) = \sum\nolimits_t \frac{1}{n^t} \sum\nolimits_i d_\phi\left(x_i^t \| u_{h^t(x_i^t)}^t\right) \tag{9}$$

$$+ \frac{\lambda}{T-1} \sum\nolimits_{t \neq s} \sum\nolimits_{z,l} \left(w_{zl}^{ts} d_\phi(u_z^t \| u_l^s) + w_{lz}^{st} d_\phi(u_l^s \| u_z^t)\right),$$

$$s.t. \ \forall\,(t,s)\,, \sum\nolimits_z w_{zl}^{ts} = \pi_l^s \,, \sum\nolimits_l w_{zl}^{ts} = \pi_z^t \,, w_{zl}^{ts} \geq 0 \,, \forall\, z, l \,.$$

Comparing with Eq. (2-5), we discard the constant $1/T$, simply use $\sum_{t \neq s}$ to denote summation on all unordered task pairs $\{t, s\}$, and use $\sum_{z,l}$ to denote summation on all ordered pairs (z,l) between cluster z of task t and cluster l of task s. We use $\mathcal{W} = \{W^{ts}\}_{(t,s)}$ to denote the set of all relation matrices.

In the problem in Eq. (9), $\pi_z^t = n_z^t / n^t$ is determined by assigning function h^t, hence the two series of equality constraints couple variables W^{ts}, h^t and h^s, which makes the problem difficult. Since the two equality constraints aim to prevent trivial solutions with very unbalanced clusters, we simply relax the problem by letting $\pi_z^t = 1/K^t$ and $\pi_l^s = 1/K^s$ be constants. This relaxed constraints can play the same role of encouraging balanced clusters, while decoupling variables.

As a general Bregman divergence is not convex w.r.t. the right variable, the relaxed problem is also non-convex. The full gradient w.r.t. all variables is complex and difficult to deal with. Fortunately, as introduced later, due to some properties of a Bregman divergence, it's easy to to alternatively optimize w.r.t. groups of variables.

Now we optimize the relaxed problem w.r.t. three groups of variables alternatively, i.e., cluster centroids $\mathcal{U} = \{\mathcal{U}^t\}_t$, assigning functions $\mathcal{H} = \{h^t\}_t$, and relation matrices $\mathcal{W} = \{W_{(z,l)}^{ts}\}$.

Optimize $\mathcal{W}$ fixing $\mathcal{U}$ and $\mathcal{H}$ Fixing $\mathcal{U}$ and $\mathcal{H}$, each matrix W^{ts} is independently determined by following problem:

$$\min_{W^{ts}} \sum\nolimits_{z=1}^{K^t} \sum\nolimits_{l=1}^{K^s} w_{zl}^{ts}\, d_{zl}^{ts} \tag{10}$$

$$s.t. \sum\nolimits_z w_{zl}^{ts} = 1/K^s \,, \sum\nolimits_l w_{zl}^{ts} = 1/K^t \,, w_{zl}^{ts} \geq 0 \,, \forall\, z, l,$$

where $d_{zl}^{ts} = d_\phi(u_z^t \| u_l^s)$ is the Bregman divergence from the centroid of cluster z of task t to that of cluster l of task s . This problem can be efficiently solved by standard linear programming techniques, such as simplex method. In fact, we need only to iterate few iterations to promise descent rather than completely iterate to a minimum.

The problem will lead to such a solution that w_{zl}^{ts} is large if d_{zl}^{ts} is small. Hence it functions as matching similar clusters among tasks and w_{zl}^{ts} measures similarity between cluster z of task t and cluster l of task s.

Optimize $\mathcal{H}$ fixing $\mathcal{U}$ and $\mathcal{W}$ Fixing $\mathcal{U}$ and $\mathcal{W}$, each assigning function h^t for task t is also independently determined by

$$\min_{h^t} \sum\nolimits_i d_\phi\left(x_i^t \| u_{h^t(x_i^t)}^t\right) . \tag{11}$$

Obviously, the optimum of this problem is

$$h^t(x_i^t) = \arg\min\nolimits_z d_\phi\left(x_i^t \| u_z^t\right) , \tag{12}$$

which makes each item within the summation in Eq. (11) is minimized.

Optimize $\mathcal{U}$ fixing $\mathcal{W}$ and $\mathcal{H}$ Fixing $\mathcal{W}$ and $\mathcal{H}$, the centroids $\mathcal{U} = \{\mathcal{U}^t\}_t$ of different tasks are correlated resulted by the task regularization. Hence we traverse all the tasks to optimize each $\mathcal{U}^t$ sequentially, with the centroids $\{\mathcal{U}^s\}_{s \neq t}$ of other tasks fixed. This is an unconstrained problem:

$$\min_{\mathcal{U}^t} \frac{1}{n^t} \sum\nolimits_i d_\phi\left(x_i^t \| u_{h^t(x_i^t)}^t\right)$$
$$+ \frac{\lambda}{T-1} \sum\nolimits_{s \neq t} \sum\nolimits_{z,l} \left(w_{zl}^{ts} d_\phi(u_z^t \| u_l^s) + w_{lz}^{st} d_\phi(u_l^s \| u_z^t)\right) .$$

In fact, if we denote $\mathcal{I}_z^t = \{i | h^t(x_i^t) = z\}$ as the set of all members of cluster z of task t, then the first item can be written as another form $\frac{1}{n^t} \sum_{z=1}^{K^t} \sum_{i \in \mathcal{I}_z^t} d_\phi(x_i^t \| u_z^t)$. Substituting this form with the first item in above problem, we can find that in task t, each centroid u_z^t in $\mathcal{U}^t$ is independently determined by

$$\min_{u_z^t} \frac{1}{n^t} \sum\nolimits_{i \in \mathcal{I}_z^t} d_\phi(x_i^t \| u_z^t) \tag{13}$$
$$+ \frac{\lambda}{T-1} \sum\nolimits_{s \neq t} \sum\nolimits_{l=1}^{K^s} \left(w_{zl}^{ts} d_\phi(u_z^t \| u_l^s) + w_{lz}^{st} d_\phi(u_l^s \| u_z^t)\right) .$$

This is the most difficult phase in the whole problem. Luckily, following two properties of a Bregman divergences make this problem easier. Let $\omega_i \geq 0$, and $\sum_i \omega_i > 0$, then

Theorem 1 $\sum_{i=1}^n \omega_i d_\phi(x_i \| \theta) = A\, d_\phi(\mu_L \| \theta) + C$, *where* $A = \sum_{i=1}^n \omega_i$, $\mu_L = \sum_i \omega_i x_i / A$, *$C$ is a constant w.r.t. θ .*

Theorem 2 $\sum_{i=1}^n \omega_i d_\phi(\theta \| x_i) = A\, d_\phi(\theta \| \mu_R) + C$, *where* $A = \sum_{i=1}^n \omega_i$, $\mu_R = (\nabla\phi)^{-1}\left(\sum_i \omega_i \nabla\phi(x_i)/A\right)$, *and C is a constant w.r.t. θ . $\nabla\phi$ is the gradient of function ϕ, and $(\nabla\phi)^{-1}$ is the inverse function of $\nabla\phi$.*

(Nielsen and Nock 2009) provide proofs to above two theorems for a special case of $\omega_i = 1/n$. In above general case of ω_i, proofs are straightforward and almost the same, hence are omitted here.

Directly applying above two theorems, we can write Eq. (13) in a concise form as

$$\min_{u_z^t} A \cdot d_\phi\left(\mu_L \| u_z^t\right) + B \cdot d_\phi\left(u_z^t \| \mu_R\right) , \tag{14}$$

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

$$\mu_L = \frac{1}{A}\left(\frac{1}{n^t} \sum\nolimits_{i \in \mathcal{I}_z^t} x_i^t + \frac{\lambda}{T-1} \sum\nolimits_{s \neq t} \sum\nolimits_{l=1}^{K^s} w_{lz}^{st} u_l^s\right)$$
$$\mu_R = (\nabla\phi)^{-1}\left(\frac{\lambda}{B \cdot (T-1)} \sum\nolimits_{s \neq t} \sum\nolimits_{l=1}^{K^s} w_{zl}^{ts} \nabla\phi(u_l^s)\right) .$$

Now the problem left is to solve Eq. (14). There are two cases:

- **When $d_\phi(x\|y)$ is symmetric,** i.e., $d_\phi(x\|y) = d_\phi(y\|x)$, such as squared Euclidean distance and Mahalanobis distance, applying Theorem 1 again, the objective of Eq. (14) is equal to $(A+B) d_\phi\left(\frac{A \cdot \mu_L + B \cdot \mu_R}{A+B} \| u_z^t\right)$. According to the nonnegative property of a Bregman divergence, a minimum is
$$u_z^t = (A \cdot \mu_L + B \cdot \mu_R) \,/\, (A+B) . \tag{15}$$
- **When $d_\phi(x\|y)$ is asymmetric,** i.e., $d_\phi(x\|y) \neq d_\phi(y\|x)$, such as KL divergence, we cannot obtain a closed form minimum. (Nielsen and Nock 2009) provided an efficient *Geodesic-walk dichotomic approximation* algorithm for this problem.

Up to now, all variables have been traversed through and optimized alternatively. The overall process is listed in Algorithm 1.

Algorithm 1 Multitask Bregman Clustering

```
Require: Data sets $\mathcal{X}$ of $T$ tasks; Clustering numbers $\mathcal{K}$ of all
    tasks; Parameter $\lambda \geq 0$; Initial clustering assignments $\mathcal{H}$ of all
    tasks.
Initialization. Initialize $\mathcal{U}$ according to $\mathcal{H}$ not considering task
    regularization.
repeat
    Learning relationship between clusters of different
    tasks: Update $\mathcal{W}$ by solving the set of linear programming
    of Eq. (10).
    Assigning each sample to a cluster: Update assigning
    functions $\mathcal{H}$ of all tasks according to Eq. (12) .
    for $t = 1$ to $T$ do
        Updating cluster centroids $\mathcal{U}^t$ for task $t$: for each clus-
        ter $z = \{1, \ldots, K^t\}$, using Eq. (15) (for a symmetric
        Bregman divergence) or the geodesic-walk dichotomic
        approximation method of (Nielsen and Nock 2009) (for
        an asymmetric Bregman divergence) to solve Eq. (14) to
        update the centroid $u_z^t$.
    end for
until Loss function $\mathfrak{L}$ does not descend significantly up to a
    specified precision.
```

Experiments

In this section, we report experiments on several benchmark data sets. The Bregman divergence we used is the Euclidean distance. The proposed method is general for multiple tasks, but for clarity of comparison, our experiments only involves two tasks.

Data sets

We use data sets in (Zhong and Ghosh 2003) [1] , as listed in Tab. 2, where n is the size of original data set, d is the

[1] The clean data in *matlab* format are available from http://www.shi-zhong.com/software/docdata.zip .

Table 2: Data sets.

				$\mathcal{X}^1$		$\mathcal{X}^2$	
	n	d	K	n^1	K^1	n^2	K^2
NG20-1	19949	43585	20	9975	20	9974	20
NG20-2	19949	43585	20	15952	16	14969	15
NG20-3	19949	43585	20	19949	20	19949	6
reviews	4069	18482	5	3657	4 (1-4)	3070	4 (2-5)
sports	8580	14866	7	7771	6 (1-6)	5168	6 (2-7)
hitech	2301	10080	6	2114	5 (1-5)	1816	5 (2-6)
ohscal	11162	11465	10	8452	8 (1-8)	9294	8 (2-9)
tr11	414	6412	9	388	7 (1-7)	324	7 (3-9)
tr23	204	5814	6	204	6 (1-6)	157	6 (1-6)
tr45	690	8249	10	478	7 (1-7)	440	7 (4-10)

dimensionality, K is the number of categories. More details about these data sets can be found in (Zhong and Ghosh 2003).

Different from traditional clustering, our experiments involve two tasks on two data corpora. We construct the two data corpora $\mathcal{X}^1$ and $\mathcal{X}^2$ by splitting an original data set. In general, we split a data set into two parts $\mathcal{X}^1$ and $\mathcal{X}^2$ sharing some categories. The splitting schemes are also provided in Tab. 2, where n^1 and n^2 are size of the two constructed corpora respectively, and K^1 and K^2 are cluster numbers of two constructed corpora.

Original NG20 data set is composed of 6 root categories, under which are 20 sub categories. We use three splitting schemes to construct three data sets to demonstrate three typical cases of multitask clustering. The first case is that two tasks' data sets are from a same distribution. We construct a data set NG20-1 to represent this case, by randomly splitting entire NG20 into two parts, with each part having all the 20 sub categories. The second case is that two tasks' data distributions are not identical but similar, which is represented by a data set NG20-2. In NG20-2, the first data corpus $\mathcal{X}^1$ includes the 16 sub categories under root categories "alt", "comp", "misc" , "rec", "sci" and "soc", while the second data corpus $\mathcal{X}^2$ includes the 15 sub categories under "alt", "misc" , "rec", "sci", "soc" and "talk". The third case is that two tasks are on an identical data set but requires clusters at different resolutions. A data set NG20-3 is constructed to represent this case. In NG20-3, $\mathcal{X}^1$ and $\mathcal{X}^2$ are both the original NG20. However, we want to partition $\mathcal{X}^1$ into $K^1 = 20$ clusters (the 20 sub categories) and partition $\mathcal{X}^2$ into $K^2 = 6$ clusters (the 6 root categories).

All remaining data sets are used to construct data sets for the general second case. We split each data set into two parts sharing categories. For example, for data set "ohscal" in Tab. 2 the items "8 (1-8)" under column K^1 and "8 (2-9)" under column K^2 mean that $\mathcal{X}^1$ includes original categories 1 to 8, and $\mathcal{X}^2$ includes original categories 2 to 9. The representations are the same for other data sets.

Settings

We compare multitask Bregman clustering with the traditional isolated method, i.e., performing Bregman clustering (Banerjee et al. 2005) (*k-means* in this experiment) independently on each task, denoted by "IND". Our approach is called "MBC".

Both methods have the problem of local optimum and rely on initializations. As a result, we run both methods for $N = 100$ times and compare the average performance. In each run, on each task, the two methods are enforced with an identical random initialization. After each run, we obtain performance on above evaluation metrics. Parameter λ is set to 0.5 for all data sets.

Evaluations

We evaluate the performance from two aspects. The first is clustering quality of each task, evaluated by two widely used metrics *Normalized Mutual Information* (NMI) and *Adjusted Random Index* (ARI) (Willigan and Cooper 1986). Higher values on NMI and ARI means better coherence between clustering assignments and true category labels. The second is coherence among clustering results of different tasks. Two metrics are used. One is EMD between the two partitions of two tasks, as defined by Eq. (8). We denoted it as d_M. Another is NMI between the two partitions, which evaluates the coherence between two partitions on a same data set. We denote it as s_{nmi}. Notice that this metric is only used on NG20-3, where the two data corpora are identical and the metric makes sense. A lower value on d_M and a higher value on s_{nmi} mean better coherence between the two clustering results of two tasks.

Results

The mean and standard deviation values of NMI and ARI are listed in Tab. 3, from which we can see that MBC indeed significantly improve the clustering performance of each task.

The mean and standard deviation values of d_M and s_{nmi} are listed out in Tab. 4. In the table, the "true" column is the EMD between the two partitions defined by the true category labels of two tasks. It reflects intrinsic divergence between the true cluster structures of two tasks. Since the d_M itself appears as a part in objective function of the MBC, it's not strange that d_M of MBC is generally low. However, from the table, we observe an interesting result, i.e., the "true" value of d_M is indeed the lowest. It means that when the distributions of two data corpora are similar, the objective of minimizing the divergences between two models is coherent with the isolated clustering objective of each task. This phenomenon validates that collaboratively performing multiple similar clustering tasks is indeed expected to improve the clustering performance of each task. In addition, on NG20-3, two tasks are clustering a same data set at different resolution. Much higher value of s_{nmi} indicates that MBC leads to results more coherent between two resolutions.

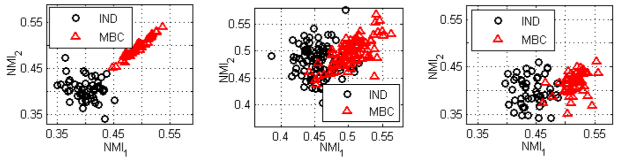
The "collaborative" property of multitask Bregman clustering are also demonstrated in Fig. 1 and Fig. 2. For each data set, after each run, we have a pair of NMI values $(\text{NMI}_1, \text{NMI}_2)$, as a point in two-dimensional space. Then after N run, we obtain N vectors of $(\text{NMI}_1, \text{NMI}_2)$ for each algorithm, and we can plotted out these points, as illustrated in Fig. 1. The figures for ARI pairs in Fig. 2 are plotted out in the same way. These figures clearly depict out the performance comparison and correlation between two methods. From the figures, it's clear that the average performance of MBC is much better than IND. Meanwhile, for IND, the two variable NMI_1 and NMI_2 seems independently, while for MBC, there is strong positive correlation between NMI_1 and NMI_2. Results for ARI pairs are similar. Limited by

Table 3: Results of NMI and ARI values $(mean \pm std)$

	Tasks	NMI		ARI	
		IND	MBC	IND	MBC
NG20-1	$\mathcal{X}^1$	$(.40 \pm .02)$	$(\mathbf{.49 \pm .02})$	$(.13 \pm .03)$	$(\mathbf{.32 \pm .02})$
	$\mathcal{X}^2$	$(.40 \pm .02)$	$(\mathbf{.49 \pm .02})$	$(.13 \pm .03)$	$(\mathbf{.33 \pm .02})$
NG20-2	$\mathcal{X}^1$	$(.46 \pm .02)$	$(\mathbf{.50 \pm .02})$	$(.19 \pm .04)$	$(\mathbf{.33 \pm .03})$
	$\mathcal{X}^2$	$(.48 \pm .03)$	$(\mathbf{.50 \pm .02})$	$(.19 \pm .05)$	$(\mathbf{.30 \pm .04})$
NG20-3	$\mathcal{X}^1$	$(.44 \pm .02)$	$(\mathbf{.52 \pm .02})$	$(.15 \pm .04)$	$(\mathbf{.35 \pm .03})$
	$\mathcal{X}^2$	$(.40 \pm .03)$	$(\mathbf{.41 \pm .02})$	$(.12 \pm .03)$	$(\mathbf{.14 \pm .02})$
reviews	$\mathcal{X}^1$	$(.32 \pm .11)$	$(\mathbf{.42 \pm .14})$	$(.19 \pm .15)$	$(\mathbf{.35 \pm .17})$
	$\mathcal{X}^2$	$(.32 \pm .17)$	$(\mathbf{.36 \pm .15})$	$(.25 \pm .22)$	$(\mathbf{.31 \pm .21})$
sports	$\mathcal{X}^1$	$(.38 \pm .08)$	$(\mathbf{.52 \pm .08})$	$(.21 \pm .10)$	$(\mathbf{.51 \pm .14})$
	$\mathcal{X}^2$	$(.46 \pm .08)$	$(\mathbf{.53 \pm .07})$	$(.30 \pm .10)$	$(\mathbf{.37 \pm .09})$
hitech	$\mathcal{X}^1$	$(.28 \pm .04)$	$(\mathbf{.29 \pm .04})$	$(.20 \pm .04)$	$(\mathbf{.26 \pm .05})$
	$\mathcal{X}^2$	$(.14 \pm .04)$	$(\mathbf{.18 \pm .04})$	$(.07 \pm .04)$	$(\mathbf{.12 \pm .04})$
ohscal	$\mathcal{X}^1$	$(.42 \pm .01)$	$(\mathbf{.43 \pm .01})$	$(.31 \pm .02)$	$(\mathbf{.32 \pm .02})$
	$\mathcal{X}^2$	$(.45 \pm .02)$	$(\mathbf{.45 \pm .02})$	$(.32 \pm .03)$	$(\mathbf{.34 \pm .03})$
tr11	$\mathcal{X}^1$	$(.51 \pm .07)$	$(\mathbf{.55 \pm .05})$	$(.41 \pm .12)$	$(\mathbf{.45 \pm .07})$
	$\mathcal{X}^2$	$(.46 \pm .06)$	$(\mathbf{.56 \pm .05})$	$(.36 \pm .10)$	$(\mathbf{.47 \pm .08})$
tr23	$\mathcal{X}^1$	$(.24 \pm .06)$	$(\mathbf{.26 \pm .07})$	$(.11 \pm .07)$	$(\mathbf{.13 \pm .08})$
	$\mathcal{X}^2$	$(.24 \pm .10)$	$(\mathbf{.26 \pm .12})$	$(.14 \pm .11)$	$(\mathbf{.16 \pm .15})$
tr45	$\mathcal{X}^1$	$(.55 \pm .06)$	$(\mathbf{.56 \pm .08})$	$(.46 \pm .10)$	$(\mathbf{.49 \pm .10})$
	$\mathcal{X}^2$	$(.49 \pm .06)$	$(\mathbf{.53 \pm .08})$	$(.32 \pm .09)$	$(\mathbf{.41 \pm .12})$

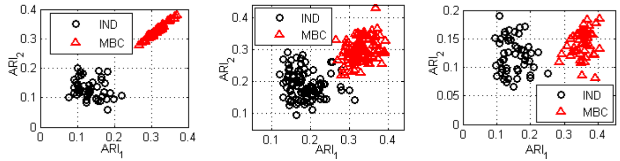
Table 4: Results: the divergences / coherence between two mixture models $(mean \pm std)$

	d_M		
	IND	MBC	true
NG20-1	(27.39 ± 5.53)	$(\mathbf{4.44 \pm 0.64})$	3.67
NG20-2	(28.61 ± 3.21)	$(\mathbf{7.92 \pm 2.87})$	6.18
reviews	(7.33 ± 2.97)	$(\mathbf{3.18 \pm 4.18})$	1.64
sports	(8.14 ± 0.77)	$(\mathbf{5.91 \pm 0.60})$	5.18
hitech	(15.97 ± 0.45)	$(\mathbf{7.42 \pm 0.23})$	2.54
ohscal	(8.57 ± 1.06)	$(\mathbf{7.67 \pm 1.01})$	4.90
tr11	(6.33 ± 0.70)	$(\mathbf{2.84 \pm 0.40})$	4.16
tr23	(8.49 ± 0.67)	$(\mathbf{8.07 \pm 0.80})$	3.55
tr45	(11.56 ± 1.65)	$(\mathbf{9.34 \pm 0.68})$	7.52
	s_{nmi}		
NG20-3	(0.47 ± 0.03)	$(\mathbf{0.67 \pm 0.01})$	



(a) NG20-1 (b) NG20-2 (c) NG20-3

Figure 1: Pairs of $(\mathrm{NMI}_1, \mathrm{NMI}_2)$ on NG20-1, NG20-2, and NG20-3 data sets.



(a) NG20-1 (b) NG20-2 (c) NG20-3

Figure 2: Pairs of $(\mathrm{ARI}_1, \mathrm{ARI}_2)$ on NG20-1, NG20-2, and NG20-3 data sets.

space, we only provide figures for three data sets, NG20-1, NG20-2 and NG20-3.

Conclusion

In this paper, we deal with multitask clustering, which aims to improve performance on each single task and also discover the relationship between clusters of different tasks. We propose a general framework and also suggest a specified approach. In our approach, we alternatively update clusters and learn relationship between clusters of different tasks, and the two phases boost each other. Based on the general Bregman divergences, our approach provides an uniform solution to a large family of data distributions and divergence assumptions, hence can be widely utilized. The approach is validated by experiments on several real data sets.

References

Ando, R., 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.; Micchelli, C. A.; Pontil, M.; and Ying, Y. 2008. A spectral regularization framework for multi-task structure learning. In *NIPS*.

Banerjee, A.; Merugu, S.; Dhillon, I. S.; and Ghosh, J. 2005. Clustering with Bregman divergences. *Journal of Machine Learning Research* 6:1705–1749.

Bregman, L. 1967. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics* 7:200–217.

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

Chakrabarti, D.; Kumar, R.; and Tomkins, A. 2006. Evolutionary clustering. In *KDD*.

Dai, W.; Yang, Q.; Xue, G.-R.; and Yu, Y. 2008. Selft-taught clustering. In *ICML*.

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

Jain, A., and Dubes, R. 1988. *Algorithms for clustering data*. Prentice Hall.

Ng, A.; Jordan, M.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *NIPS*.

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

Pan, S. J., and Yang, Q. 2008. A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China.

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

Stephens, M.; Smith, N.; and Donnelly, P. 2001. A new statistical method for haplotype reconstruction from population data. *The American Journal of Human Genetics* 68(4):978–989.

Strehl, A., and Ghosh, J. 2003. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3:583–617.

Topchy, A.; Jain, A.; and Punch, W. 2005. Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(12):1866–1881.

Willigan, G., and Cooper, M. 1986. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research* 21:441–458.

Zhang, J.; Song, Y.; Chen, G.; and Zhang, C. 2009. On-line evolutionary exponential family mixture. In *IJCAI*.

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