2012

Learning Bregman distance functions for semi-supervised clustering

Lei Wu

Chu Hong HOI
Singapore Management University, chhoi@smu.edu.sg

Rong Jin

Jianke Zhu

N. Yu

Citation

Wu, Lei; HOI, Chu Hong; Jin, Rong; Zhu, Jianke; and Yu, N.. Learning Bregman distance functions for semi-supervised clustering. (2012). *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. 24, (3), 478-491.

Available at: https://ink.library.smu.edu.sg/sis_research/2282

Learning Bregman Distance Functions for Semi-Supervised Clustering

Lei Wu, Steven C.H. Hoi, Rong Jin, Jianke Zhu, Nenghai Yu

Abstract

Learning distance functions with side information plays a key role in many data mining applications. Conventional distance metric learning approaches often assume the target distance function is represented in some form of Mahalanobis distance. These approaches usually work well when data is in low dimensionality, but often become computationally expensive or even infeasible when handling high-dimensional data. In this paper, we propose a novel scheme of learning nonlinear distance functions with side information. It aims to learn a Bregman distance function using a non-parametric approach that is similar to Support Vector Machines. We emphasize that the proposed scheme is more general than the conventional approach for distance metric learning, and is able to handle high-dimensional data efficiently. We verify the efficacy of the proposed distance learning method with extensive experiments on semi-supervised clustering. The comparison with state-of-the-art approaches for learning distance functions with side information reveals clear advantages of the proposed technique.

Index Terms

Bregman distance, distance functions, metric learning, convex functions

I. Introduction

Effective distance function plays a key role in many machine learning and data mining techniques, such as clustering [28], [6], classification [36], regression [37], and ranking [11].

Corresponding author: Steven C.H. Hoi is with Nanyang Technological University, Singapore. E-mail: chhoi@ntu.edu.sg

Lei Wu and Nenghai Yu are with the University of Science and Technology of China, PR China.

Rong Jin is with Michigan State University, US.

Jianke Zhu is with Chinese University of Hong Kong, Hong Kong.

This work was done when Lei Wu was a research assistant at Nanyang Technological University.

Learning effective distance functions is a fundamental problem in many areas in computer science and engineering, and has a significant impact on a broad range of applications, including text data mining, information retrieval [42], content-based image and video retrieval [17], [15], [16], handwriting recognition, object recognition, tagging recommendation [39], social network analysis, gene expression analysis [23], and outlier detection, among others.

For many data mining applications, the most common approach for distance measure is Euclidian distance, which is a special case in the family of Mahalanobis distances [27], [26] with the metric matrix A set to be an identity matrix. Despite its prevalent usages, Euclidean distance is often not the best distance function for many real situations. In particular, methods that adopt Euclidian distance often assume that all variables are uncorrelated, the variance across all dimensions is one and the covariances among all variables are zeros, a scenario that is hardly achieved in real world. In addition to Euclidean distance, cosine similarity is another popular function for distance/similarity measure, especially for text mining applications. Although it has been shown effective for text retrieval applications, similar to Euclidean distance, cosine similarity assumes equal weight for every dimension, which significantly limits its applications. Besides, there are also some other distances for specific uses, such as Kullback-Leibler (KL) distance in measuring correlation between two distributions [9] and the Jensen-Shannon (JS) divergence which is adopted in measuring the concept distance in visual domain [38].

Recently, learning distance functions from data has been actively studied in data mining and machine learning. Instead of simply adopting the standard Euclidean distance, researchers have attempted to learn distance functions from data automatically [40]. Among various studies of learning distance functions, one emerging group of studies [40], [3], [17] focus on exploiting side information that is often provided in the form of pairwise constraints (i.e., a constraint indicates whether two data points are similar or not). Different from explicit class labels used for training regular classification models, the side information of pairwise constraints often can be obtained in a much easier and less expensive way. For multimedia applications, side information can be obtained from user relevance feedback log data [18], [31]. For text data mining applications, side information of pairwise constrains are available in various forms. For example, one can implicitly acquire the pairwise constraints by computing links between web documents. In addition, side information can also be obtained by mining the search engine query logs [32], which are vastly available in most commercial search engines.

In literature, various techniques have been proposed for learning distance functions from side information. Example studies in this category include the distance metric learning (DML) method that poses metric learning as a convex optimization [40], Relevant Components Analysis (RCA) [3], Discriminative Component Analysis (DCA)[17], regularized distance metric learning [31], metric propagation [43], etc. Most of these existing studies assume the distance function is represented in the form of Mahalanobis distance, which becomes hard or even infeasible to solve when handling the high-dimensional data (since the number of variables in the optimization is in the order of $O(d^2)$ where d is the dimensionality). In addition, the final learned metric space is essentially a linear transformation of the Euclidian space, which may limit the capacity for distance measure.

In contrast to the conventional DML approaches that are often restricted to learning the family of Mahalanobis distances, in this paper, we propose a novel scheme of learning Bregman distance functions with side information. We first give the formal definition of Bregman distance.

Definition 1: Let φ be a continuous-differentiable real-valued and strict convex function defined on a closed convex set Ω, a Bregman distance function $d_\varphi(x, y)$ associated with the function φ is defined as follows:

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

for any points $x, y \in \Omega$.

Bregman distance or Bregman divergence [5] enjoys several salient properties [1] for distance measure. Firstly, it generalizes Mahalanobis distance to a wider class of distance functions that could be nonlinear in terms of the input patterns. Note that the Mahalanobis distance function can be viewed as a special case of Bregman distances when choosing some quadratic convex function. Second, there is a strong connection between Bregman distances and exponential families of distributions [2]. For example, regular Kullback-Leibler divergence can be shown as a special Bregman distance when choosing the negative entropy as the convex function.

Therefore, by learning the Bregman distances, we are able to generalize the target distance function to a large family of nonlinear distance functions by choosing any feasible convex functions in order to better model real complicated patterns. Bregman distance or Bregman divergence is named after L. M. Bregman, who introduced the concept in 1967. More recently researchers in geometric algorithms have shown that many important algorithms can be general-

ized from Euclidean metrics to distances defined by Bregman divergence[2]. Although it has been explored in some previous studies [2], to the best of our knowledge, there is no existing work that directly addresses the challenge of learning Bregman distances from side information. In this paper, we formally formulate this challenging problem and present some effective algorithm with application to semi-supervised clustering with side information.

As a summary, the main contributions of this paper include: (1) we propose a novel scheme of learning Bregman distance functions with side information; (2) we formulate the bregman distance learning problem into quadratic program and present an effective algorithm that can learn the optimal Bregman function from large amount of side information efficiently; (3) we improved the clustering schemes with the Bregman distance functions learned by our technique; and (4) we then compared the performance of the improved clustering methods with other state-of-the-art clustering methods on semi-supervised clustering applications.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 proposes the framework of learning Bregman distance functions with side information, and then formulates the learning task as a quadratic optimization problem that is solved by a simple yet effective subgradient descent algorithm. Section 4 investigates the application of the Bregman distance function technique to semi-supervised clustering. Section 5 conducts extensive experiments by comparing the proposed algorithms with a number of competing distance learning algorithms for clustering tasks. Section 6 concludes this paper.

II. Related Work

Most existing studies on learning distance functions can be generally classified into two categories: unsupervised learning and supervised learning. Unsupervised learning methods do not use any training data, which usually exploit underlying data distribution or manifold structures. Examples in the unsupervised learning category include the well-known methods such as Principal Component Analysis (PCA) and Multidimensional Scaling (MDS)[29], and some manifold learning methods, such as Locally Linear Embedding (LLE)[29], Isomap [33], and Neighborhood Preserving Embedding (NPE)[14], etc.

Supervised learning approaches learn distance functions by exploring some collections of training data. These methods can be further classified into two groups.

One group of studies focus on learning distance metrics/functions from training data with

explicit class labels. Example techniques include Fisher Linear Discriminant Analysis (LDA) [10] and some recently proposed methods, such as Neighbourhood Components Analysis (NCA) [19], Maximally Collapsing Metric Learning [12], metric learning for Large Margin Nearest Neighbor classification (LMNN) [36], and Local Distance Metric Learning [41], etc. The other group of studies focus on learning distance metrics from training data that are present in the form of pairwise constraints. Our work is closely related to this group of studies. We briefly introduce some background and several representative studies below.

Most existing studies aim to learn some distance metric in the form of Mahalanobis distances. In particular, for any two data points given in a vector space $x_a, x_b \in \mathbb{R}^D$, let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be the distance metric, the formula of Mahalanobis distance measure can be expressed as follows:

$$d_{\mathbf{A}}(x_a, x_b) = \|x_a - x_b\|_{\mathbf{A}} = \sqrt{(x_a - x_b)^\top \mathbf{A}(x_a - x_b)}\,, \tag{1}$$

where $\mathbf{A}$ is a symmetric and positive semi-definite matrix that parameterizes a family of Mahalanobis distances.

A variety of techniques have been proposed for learning an optimal metric in the above framework from training data that are given in the forms of pairwise constraints. In [40], the authors proposed a Probabilistic Global Distance Metric Learning (PGDM) method by formulating it into a convex optimization task and then solving it by an iterative projection algorithm. In [3], the authors proposed a Relevant Component Analysis (RCA) technique that exploits only similar pairwise constraints for metric learning. The basic idea of RCA is to identify and down-scale global unwanted variability of the data. In particular, given pairwise constraints, RCA first forms a set of "chunklets", each of them is defined as a group of data points linked together by similar (positive) pairwise constraints. The optimal Mahalanobis matrix learned by RCA is then computed as the inverse of the average covariance matrix of the chunklets. RCA enjoys the merit of simplicity, but has the limitation of ignoring the dissimilar (negative) constraints. In [17], the authors proposed an extension of RCA, known as Discriminative Component Analysis (DCA), to explicitly address such limitation. To find the optimal metric, DCA learns the optimized transformation by both maximizing the total variance between the discriminative chunklets and minimizing the total variance of data points in the same chunklets. Recently, the authors in [8] proposed an Information-Theoretic Metric Learning (ITML) approach that also learns the Mahalanobis distance.

Besides, there is also some work for addressing the robustness of metric learning. In [31], the authors proposed a regularized distance metric learning method to improve the work in [40] for noisy data. Very recently semi-supervised distance metric learning also has been proposed for improving the metric robustness by exploiting both side information and unlabeled data information, which basically follows the typical framework of learning Mahalanobis distances.

Different from the existing studies above, the proposed technique in this paper goes beyond the conventional distance metric learning framework, which is often limited for learning only the family of Mahalanobis distances. Lastly, we notice that there is some existing work on studying Bregman divergence for data clustering [2]. Our work is considerably different from the previous study [2] in two folds. Firstly, the previous work aims to directly optimize the clustering performance by using Bregman divergence, while we aim to learn general distance functions, which can be applied in a broader range of applications. Secondly, their approach is in general an unsupervised method that does not exploit side information, while our work aims to learn with side information.

III. Learning Bregman Distance Functions

In this section, we formally formulate the problem of learning Bregman distance functions from side information. To make our discussion more readable, Table 1 lists the symbols to be frequently used in the following discussions.

A. Problem Formulation

In general, we aim to learn a Bregman distance function using a non-parametric approach that is similar to Support Vector Machines (SVM) [7]. In particular, given a reproducing kernel Hilbert space denoted by $\mathcal{H}_\kappa$, we aim to search for a convex function $\varphi(x) \in \mathcal{H}_\kappa$ such that the induced Bregman distance function, denoted as $d_\varphi(x_a, x_b)$, minimizes the overall training error.

We denote by $\mathcal{D} = \{(x_a^k, x_b^k, y_k), k = 1, \ldots, n\}$ the collection of training instances, where n is the number of pairs. Each training instance is a tuple consisting of three elements: x_a^k and x_b^k are two examples represented by vectors in a d-dimensional space; y_k is a class label assigned to the pair of examples x_a^k and x_b^k. The class label is positive, i.e., $y_k = +1$ when x_a^k and x_b^k are in the same class, and negative $y_k = -1$ otherwise. We also introduce $X = (x_1, \ldots, x_N)$ to represent all the input training instances in $\mathcal{D}$, where N is the number of samples.

Symbol	Meaning
κ	kernel function
φ	convex function
$\mathcal{H}$	functional space
$\mathcal{H}_\kappa$	functional space with representer kernel κ
Ω	feature space
$\Omega_\mathcal{H}$	a convex subspace of functional space $\mathcal{H}$
$\mathcal{L}$	loss function
ℓ	hinge loss
$\mathcal{D}$	collection of data samples
S_t^+	training set for the t-th iteration
x	data sample
x_a, x_b	sample pair
X	sample matrix
y	label of a sample pair
n	number of pairwise constraints
N	number of samples
A	Mahanalobis distance metric
d_A	Mahanalobis distance of metric A
d_B	Symmetric Bregman distance function
d_φ	Asymmetric Bregman distance function
K	$N \times N$ kernel matrix
k	the k-th pair-wise constraint
t	the t-th iteration
C	penalty cost parameter
I	Identity matrix
β	threshold parameter
γ	learning rate
D	dimensionality

TABLE I

THE NOTATION TABLE OF FREQUENTLY USED SYMBOLS

To define the Bregman distance function, we need to first define a convex function $\varphi(x) : \mathbb{R}^D \mapsto \mathbb{R}$. We assume $\varphi(x)$ is a strict convex function and is twice differentiable. Since a general Bregman distance function is asymmetric and is therefore not a metric, we define the following symmetric Bregman distance function

$$d_B(x_a, x_b) = (\nabla\varphi(x_a) - \nabla\varphi(x_b))^\top (x_a - x_b) \tag{2}$$

which can be considered as the approximation of the asymmetric Bregman distance by Taylor expansion. The following proposition indicates the properties of $d_B(x_a, x_b)$.

Proposition 1: The distance function defined in (2) satisfies the following properties if $\varphi(x)$ is a strictly convex function: (a) $d_B(x_a, x_b) = d_B(x_b, x_a)$, (b) $d_B(x_a, x_b) \geq 0$, (c) $d_B(x_a, x_b) = 0 \leftrightarrow x_a = x_b$.

Proof: The first property (a) follows directly from the fact $d_B(x_a, x_b) = d_\varphi(x_a, x_b) + d_\varphi(x_b, x_a)$ where $d_\varphi(x_a, x_b)$ is an asymmetric Bregmen distance function that is defined as

$$d_\varphi(x_a, x_b) = \varphi(x_a) - \varphi(x_b) - \nabla\varphi(x_b)^\top (x_a - x_b).$$

Since $\varphi(x)$ is strictly convex, we obtain $d_\varphi(x_a, x_b) > 0$, and thus have the second property (b).

The third property (c) follows the following fact:

$$d_B(x_a, x_b) = (x_a - x_b)^\top \nabla^2\varphi(\tilde{x})(x_a - x_b)$$

where $\tilde{x}$ is a point lying between x_a and x_b. Given $\varphi(x)$ is a strictly convex function, we have $\nabla^2\varphi(\tilde{x}) \succ 0$, which leads to the third property. The Hessian matrix form can be justified by the Mean value theorem [21]. ■

The Bregman distance function can be viewed as a general Mahalanobis distance that introduces a local distance metric $A = \nabla^2\varphi(\tilde{x})$. Unlike the conventional Mahalanbis distance where metric A is a constant matrix throughout the entire space, the local distance metric $A = \nabla^2\varphi(\tilde{x})$ is introduced via the Hessian matrix of convex function $\varphi(x)$ and therefore depends on the location of x_a and x_b.

Although the Bregman distance function defined in (2) does not satisfy the triangle inequality, the following proposition shows the degree of violation could be bounded if the Hessian matrix of $\varphi(x)$ is bounded.

Proposition 2: Let Ω be the closed domain for x. If $\exists m, M \in \mathbb{R}$, $M > m \geq 0$ and

$$mI \preceq \nabla^2\varphi(x) \preceq MI, \ \forall x \in \Omega$$

where I is the identity matrix, we have the following inequality

$$\sqrt{d_B(x_a, x_b)} \leq \sqrt{d_B(x_a, x_c)} + \sqrt{d_B(x_c, x_b)} \quad (3)$$

$$+ \ (\sqrt{M} - \sqrt{m})[d_B(x_a, x_c) d_B(x_c, x_b)]^{1/4} \quad (4)$$

The details of our proof for the above proposition can be found in Appendix A [1]. As indicated in Proposition 2, the degree of violation of the triangle inequality is essentially controlled by $\sqrt{M} - \sqrt{m}$. Given a smooth convex function with almost constant Hessian matrix, we would expect that to a large degree, Bregman distance will satisfy the triangle inequality. In the extreme case when $\varphi(x) = x^\top Ax/2$ and $\nabla^2\varphi(x) = A$, we have a constant Hessian matrix, leading to a complete satisfaction of the triangle inequality.

Following the maximum margin framework, a straightforward approach is to consider the following optimization problem, i.e.,

$$\min_{\varphi\in\Omega_{\mathcal{H}_\kappa},\beta\in\mathbb{R}} \quad \frac{1}{2}|\varphi|^2_{\mathcal{H}_\kappa} + C\sum_{k=1}^{n}\ell(y_k[d_B(x_a^k, x_b^k) - \beta]) \tag{5}$$

where $\Omega_{\mathcal{H}} = \{f \in \mathcal{H} : f \text{ is convex}\}$ refers to the subspace of functional space $\mathcal{H}$ that only includes convex functions, $\ell(z) = \max(0, 1-z)$ is a hinge loss, and C is a penalty cost parameter.

The main issue with the variational problem in (5) is that it is difficult to derive a representer theorem for $\varphi(x)$ because it is $\nabla\varphi(x)$ used in the definition of distance function rather than $\varphi(x)$. Below we will consider a special family of kernel functions $\kappa(x_a, x_b)$ that allows for the derivation of the representer theorem.

In this paper, we consider $\kappa(x_a, x_b)$ in the following special forms

$$\kappa(x_a, x_b) = h(x_a^\top x_b) \tag{6}$$

where $h : \mathbb{R} \mapsto \mathbb{R}$ is a univariate convex function. Note that not all the convex function $h(\cdot)$ will make $\kappa(\cdot,\cdot)$ defined (6) satisfy the Mercer's condition (i.e., a kernel function). Examples of $h(z)$ that makes a kernel function $\kappa(\cdot,\cdot)$ in (6) are $h(z) = z^i$ or $h(z) = (z+1)^i$ (i.e., the polynomial kernel). It is also not difficult to show that $h(z) = \exp(z)$ will result in a kernel function for $\kappa(\cdot,\cdot)$.

We then consider how to restrict $h(z)$ to make the derivation of the representer theorem for (5) easy. Without loss of generality, we assume $h(0) = 0$. First, since $\varphi(x) \in \mathcal{H}_\kappa$, we have

$$\varphi(x) = \int d\xi\kappa(x,\xi)q(\xi) = \int d\xi h(x^\top\xi)q(\xi) \tag{7}$$

where $q(\xi)$ is a weight function. We further define space $\mathcal{A}$ and $\bar{\mathcal{A}}$ as

$$\mathcal{A} = \text{span}\{x_1, \ldots, x_N\},\ \bar{\mathcal{A}} = \text{Null}(x_1, \ldots, x_N) \tag{8}$$

[1] http://www.cais.ntu.edu.sg/~chhoi/bregman/appendix.pdf

We then divide the space $\mathcal{H}_\kappa$ into $\mathcal{H}_\parallel$ and $\mathcal{H}_\perp$ that are defined as follows

$$\mathcal{H}_\parallel = \text{span}\{\kappa(x,\cdot), \forall x \in \mathcal{A}\},\ \mathcal{H}_\perp = \text{span}\{\kappa(x,\cdot), \forall x \in \bar{\mathcal{A}}\} \tag{9}$$

The following proposition summarizes an important property of the reproducing kernel Hilbert space $\mathcal{H}_\kappa$ when kernel function $\kappa(\cdot,\cdot)$ is restricted to Eq. (6).

Proposition 3: If the kernel function $\kappa(\cdot,\cdot)$ is written in the form of Equation (6) with $h(0) = 0$, we have $\mathcal{H}_\parallel$ and $\mathcal{H}_\perp$ form a complete partition of $\mathcal{H}_\kappa$, i.e., $\mathcal{H}_\kappa = \mathcal{H}_\parallel \cup \mathcal{H}_\perp$, and $\mathcal{H}_\parallel \perp \mathcal{H}_\perp$.

It is straightforward to verify the above Proposition. Using this proposition, we have the following representer theorem for $\varphi(x)$ that minimizes (5).

Theorem 1: The function $\varphi(x)$ that minimizes (5) should be $\varphi(x) \in \mathcal{H}_\parallel$. Hence, the minimizer $\varphi(x)$ is expressed as

$$\varphi(x) \in \mathcal{H}_\parallel = \int_{\xi \in \mathcal{A}} dx i q(\xi) h(x^\top \xi) = \int du q(u) h(x^\top X u) \tag{10}$$

where $u \in \mathbb{R}^N$ and $X = (x_1, \ldots, x_N)$.

The detailed proof to the above theorem can be found in the Appendix B [2].

B. Algorithm

To obtain the expression for $\varphi(x)$, we consider the following special cases for $q(\xi)$. $q(\xi) = \sum_{i=1}^N \alpha_i \delta(\xi - x_i)$ where $\alpha_i \geq 0$. This results in

$$\varphi(x) = \sum_{i=1}^N \alpha_i h(x_i^\top x), \tag{11}$$

and consequently $d_B(x_a, x_b)$ can be expressed as follows:

$$d_B(x_a, x_b) = \sum_{i=1}^N \alpha_i (h'(x_a^\top x_i) - h'(x_b^\top x_i)) x_i^\top (x_a - x_b)$$

We define $h(x_a) = (h'(x_a^\top x_1), \ldots, h'(x_a^\top x_N))^\top$, and thus we have $d_B(x_a, x_b)$ expressed as:

$$d_B(x_a, x_b) = (x_a - x_b)^\top X(\alpha \circ [h(x_a) - h(x_b)])$$

Notice that when $h(z) = z^2/2$, we have $d_B(x_a, x_b)$ expressed as

$$d_B(x_a, x_b) = (x_a - x_b)^\top X \text{diag}(\alpha) X^\top (x_a - x_b)$$

[2] http://www.cais.ntu.edu.sg/~chhoi/bregman/appendix.pdf

This is clearly related to Mahanabolis distance with metric equal to $X\text{diag}(\alpha)X^\top = \sum_{i=1}^N \alpha_i x_i x_i^\top$. When $h(z) = \exp(z)$, we have $h(x) = (\exp(x^\top x_1), \ldots, \exp(x^\top x_N))$, and the resulting distance function is no longer stationary due to the non-linear function $\exp(z)$.

In this paper, we will focus on the first situation, i.e. $q(y) = \sum_{i=1}^N \alpha_i \delta(y - x_i)$. We thus have (5) simplified as

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^N,\beta} \quad & \frac{1}{2}\alpha^\top K\alpha + C\sum_{k=1}^n \varepsilon_k \\ \text{s. t.} \quad & y_k\left((x_a^k - x_b^k)^\top X(\alpha \circ [h(x_a^k) - h(x_b^k)]) - \beta\right) \geq 1 - \varepsilon_k, \\ & \varepsilon_k \geq 0,\ k = 1, \ldots, n, \alpha_i \geq 0, i = 1, \ldots, N \end{aligned} \tag{12}$$

Note that the constraint $\alpha_i \geq 0$ is introduced to ensure $\varphi(x) = \sum_{i=1}^N \alpha_i h(x^\top x_i)$ is a convex function. Here we assume $h(z)$ is a convex function. For the convenience of presentation, we introduce the notation z_k as

$$z_k = [h(x_a^k) - h(x_b^k)] \circ [X^\top(x_a^k - x_b^k)] \tag{13}$$

Then, the problem in (12) becomes

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^N,\beta} \quad & \mathcal{L} = \frac{1}{2}\alpha^\top K\alpha + C\sum_{k=1}^n \ell(y_k[z_k^\top\alpha - \beta]) \\ \text{s. t.} \quad & \alpha_i \geq 0, i = 1, \ldots, N \end{aligned} \tag{14}$$

where $\ell(z) = \max(0, 1 - z)$. We can solve the above problem by a simple subgradient descent approach. In particular, at each iteration, given solution α and β, we compute the gradients as

$$\nabla_\alpha \mathcal{L} = K\alpha + C\sum_{k=1}^n \partial\ell(y_k[z_k^\top\alpha - \beta])y_k z_k, \tag{15}$$

$$\nabla_\beta \mathcal{L} = -C\sum_{k=1}^n \partial\ell(y_k[z_k^\top\alpha - \beta])y_k \tag{16}$$

Let us denote by $\mathcal{S}_t^+ \in \mathcal{D}$ as the set of training examples for which (α^t, β^t) suffers a non-zeros loss, i.e. $\mathcal{S}_t^+ = \{(z_k, y_k) \in \mathcal{D} : y_k(z_k^\top\alpha - \beta) < 1\}$. We can then express the sub-gradients of $f(\alpha, \beta; \mathcal{D})$ at α and β as follows:

$$\nabla_\alpha \mathcal{L} = K\alpha - C\sum_{(z_k,y_k)\in\mathcal{S}_t^+} y_k z_k, \tag{17}$$

$$\nabla_\beta \mathcal{L} = C\sum_{(z_k,y_k)\in\mathcal{S}_t^+} y_k \tag{18}$$

The new solution, denoted by α' and β', is computed as

$$\alpha'_i = \pi_{[0,+\infty]}\left(\alpha_i - \gamma[\nabla_\alpha \mathcal{L}]_i\right),\ \beta' = \beta - \gamma\nabla_\beta \mathcal{L}$$

where α'_i denotes the ith element of vector α' $\pi_G(x)$ projects x into the domain G, and γ is the stepsize that can be decided by a line search. In our approach, for the learning rate, we set it as $\gamma = \frac{C}{t}$ similar to the approach used in Pegasos algorithm [30] for solving SVM problems. The pseudo-code of the proposed algorithm is summarized in Algorithm 1.

Algorithm 1 Algorithm of Learning Bregman distance functions

INPUT:

- data matrix: $X \in \mathbb{R}^{N\times d}$
- pair-wise constraint (x_a^k, x_b^k, y^k), where x_a^k and x_b^k are the points, and constraints $y_k = \{+1, -1\}$
- penalty cost parameter C

OUTPUT:

- Bregman coefficients α, β

initialize Bregman coefficients: $\alpha = \alpha_0$, $\beta = \beta_0$

calculate kernel matrix: $\kappa(x_a, x_b) = h(x_a^\top x_b)$

calculate vectors z_k:

$z_k = [h(x_a^k) - h(x_b^k)] \circ [X^\top(x_a^k - x_b^k)]$

set iteration step t = 1;

repeat

(1) update the learning rate:

$\gamma = C/t$, t = t + 1

(2) update subset of training instances:

$\mathcal{S}_t^+ = \{(z_k, y_k) \in \mathcal{D} : y_k(z_k^\top \alpha - \beta) < 1\}$

(3) compute the gradients w.r.t α and β

$\nabla_\alpha \mathcal{L} = K\alpha - C\sum_{z_k\in\mathcal{S}_t^+} y_k z_k$,

$\nabla_b \mathcal{L} = C\sum_{z_k\in\mathcal{S}_t^+} y_k$

(4) update Bregman coefficients:

for $i = 1, 2, \ldots,$ N **do**

$\alpha_i = \pi_{[0,+\infty]}\left(\alpha_i - \gamma[\nabla_\alpha \mathcal{L}]_i\right)$

end for

$\beta' = \beta - \gamma\nabla_\beta \mathcal{L}$

until convergence

C. Complexity Analysis

Here we discuss the time complexity of the algorithm. In general, the whole algorithm for learning the bregman distance function has the worst case time complexity $\mathcal{O}(N^2d)$, where d is data dimensionality and N is the number of data points in the training data collection $\mathcal{D}$. The main computation lies in the initial steps of computing kernel matrix K and vectors z_k. If we examine the subgradient descent algorithm itself, each step essentially has a linear complexity, i.e., $\mathcal{O}(\max(N, n))$. In some real application, as matrix K and vectors z_k could be pre-computed, the proposed solution is scalable to large-scale online applications.

IV. Bregman Clustering

A. Overview

In general, the method of learning Bregman distance functions can be applied to any application that needs to find good distance functions with side information. In this section, we present an application, semi-supervised clustering, that adopts the distance function learned by our technique.

In terms of efficiency and popularity, we consider to apply the Bregman distance function in two types of well-known clustering techniques, i.e, the basic k-means clustering algorithm [13], [20] and the hierarchial clustering [22]. It is worth noting that the Bregman function can be also applied to other clustering algorithms. Due to the limit of space, we do not adopt other clustering methods here.

B. P2C Bregman k-means Algorithm

One straightforward approach is to extend the regular k-means algorithm by simply replacing the Euclidean distance with the Bregman distance. In particular, following the standard procedure of k-means algorithm, we can repeat the following two-step iteration: (1) form k clusters by assigning each data point to its closest cluster based on Bregman distances, and (2) update the centroid of each cluster.

Specifically, for each data example x_i, we can compute the Bregman distance between x_i and a cluster centroid ω_l as:

$$d_B(x_i, \omega_l) = (\nabla\varphi(x_i) - \nabla\varphi(\omega_l))^\top (x_i - \omega_l) \tag{19}$$

By choosing function $\varphi(x) = \sum_{i=1}^{N} \alpha_i h(x^\top x_i)$ and some h function, e.g. $h(z) = z^2/2$, it can be further written as:

$$d_B(x_i, \omega_l) = (\alpha \circ X^\top x_i - \alpha \circ X^\top \omega_l)(x_i - \omega_l) \tag{20}$$

where α are the Bregman coefficients that are learned by the proposed Bregman distance learning algorithm, X contains input patterns of all training instances, and ω_l denotes the lth cluster centroid. In practice, as α is usually a sparse vector (with only a few nonzero values), the Bregman distance function can be computed efficiently.

In the above clustering approach, we measure the closeness of a data point to a cluster by computing the Bregman distance from the data point to the cluster centroid. We refer to such a clustering approach as a Point-to-Centroid Bregman k-means clustering method (BKM-P2C).

C. P2P Bregman k-means Algorithm

Similar to the regular k-means algorithm, the above P2C Bregman k-means algorithm represents each cluster by its centroid, which implicitly assumes data clusters are given in some spherical shape, which may not be the case in real data. To address the limitation, instead of adopting the cluster centroid, we propose another Point-to-Point Bregman k-means clustering algorithm (BKM-P2P), which does not make the spherical shape assumption. The similar ideas have been adopted in some previous clustering algorithms, such as agglomerative hierarchical clustering [4], [25].

Different from the P2C approach, the point-to-point algorithm measures the closeness of a data point x_i to a cluster ω_l by computing the average Bregman distance from data point x_i to all points in the cluster ω_l. It seems that such approach may be computationally intensive for computing many pairwise distances. However, as we can store the computed distances, it avoids the needs of frequently updating the cluster centroids during clustering. As a result, the computation of Bregman distances can be essentially reduced when performing online clustering. Thus, in terms of online clustering computation, the P2P algorithm is essentially more efficient than the P2C algorithm.

D. Bregman Distance Function with Application to Hierarchical Clustering

The scheme of learning Bregman distance functions is beneficial to hierarchical clustering, which is a clustering method that aims to build a hierarchy of clusters. In hierarchical clustering,

Algorithm 2 Point-to-Centroid Bregman k-means algorithm (BKM-P2C)

INPUT:

- data matrix: $X \in \mathbb{R}^{N \times d}$
- Bregman coefficient: $\alpha_i,\ i = 1, \cdots, n$
- number of clusters: K

OUTPUT:

- cluster partition: $\mathcal{W} = \{\mathcal{W}_l\}_{l=1}^{K}$, the center of $\mathcal{W}_l$ is ω_l

initialize cluster partition $\mathcal{W} = \{\mathcal{W}_l\}_{l=1,\cdots,K}$

set t = 0

repeat

  set $\mathcal{W}_l^{t+1} = \emptyset$, $l = 1, \ldots, K$

  for $i = 1, 2, \ldots, N$ **do**

    (1) compute P2C Bregman distances:

    $$d_B(x_i, \omega_l) = (\nabla\varphi(x_i) - \nabla\varphi(\omega_l))^\top (x_i - \omega_l)$$

    where $\nabla\varphi(x) = \sum_i \alpha_i x_i^\top x \cdot x_i$

    (2) compute cluster assignment

    $$l^* = \arg\min_l d_B(x_i, \omega_l)$$

    (3) update the cluster partition:

    $$\mathcal{W}_{l*}^{t+1} = \mathcal{W}_{l*}^{t+1} \cup \{x_i\}$$

  end for

  (4) update the centroids of clusters:

  $\omega_l = \frac{1}{|\mathcal{W}_l^{t+1}|} \sum_{x_i \in \mathcal{W}_l^{t+1}} x_i$

  t = t + 1

until convergence

data examples are organized in a hierarchical tree structure, as shown in Figure 1. Different from regular "flat" clustering approaches, such as k-means clustering, hierarchical clustering does not require to specify the number of clusters, and thus may be flexible and provide more informative results in some applications.

In general, there are two types of methods for hierarchical clustering: *agglomerative* methods, which perform clustering via a series of "bottom-up" merging steps, and *divisive* methods, which perform clustering via a series of "top-down" partitioning steps. In this study, we adopt the agglomerative hierarchical clustering methods that are more commonly used in previous studies.

Algorithm 3 Point-to-Point Bregman k-means Algorithm (BKM-P2P)

INPUT:

- data matrix: $X \in \mathbb{R}^{N \times d}$
- Bregman coefficients: α_i
- number of clusters: K

OUTPUT:

- cluster partition: $\mathcal{W}$

calculate P2P Bregman distance matrix D:

$d_B(x_a, x_b) = (\nabla\varphi(x_a) - \nabla\varphi(x_b))^\top (x_a - x_b)$

where $\nabla\varphi(x) = \sum_i \alpha_i x_i^\top x \cdot x_i$

initialize cluster partition $\mathcal{W} = \{\mathcal{W}_l^0\}_{l=1,\cdots,K}$

set t = 0

repeat

  set $\mathcal{W}_l^{t+1} = \emptyset$, $l = 1, \ldots, K$

  for $i = 1, 2, \ldots, N$ **do**

    (1) compute cluster assignment:

    $k^* = \arg\min_k \frac{1}{|\mathcal{W}_k^t|} \sum_{x_j \in \mathcal{W}_k^t} d_B(x_a, x_b)$

    (2) update the cluster partition:

    $\mathcal{W}_{l*}^{t+1} = \mathcal{W}_{l*}^{t+1} \cup \{x_i\}$

  end for

  set t = t + 1

until convergence

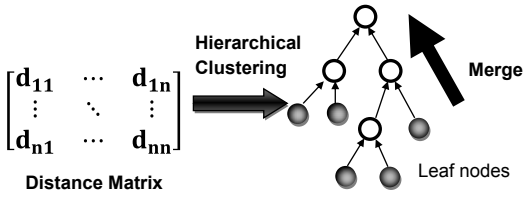


Fig. 1. Illustration of the agglomerative hierarchical clustering.

The application of our technique of learning Bregman distance functions to agglomerative hierarchical clustering is rather straightforward. In particular, a regular agglomerative clustering method [4], [25] is achieved by firstly building a linkage graph based on the similarity between samples and then group the most similar samples to form a binary tree, which is used to store

the hierarchical clusters. Hierarchical clusters can be easily transformed into flat clusters by cutting the binary tree by a threshold. A regular agglomerative clustering often simply computes the linkage graph by Euclidean distance in many applications. Our approach is to improve the agglomerative clustering by replacing the Euclidean distance with the Bregman distance function learned by the proposed scheme. The remaining clustering process is the same to the regular agglomerative clustering method.

V. EXPERIMENTS

We evaluate the performance of the proposed Bregman distance function technique with applications to semi-supervised clustering tasks. We compare our technique with a number of competing distance metric learning algorithms. These algorithms are the state-of-the-art distance metric learning approaches in machine learning.

A. Experimental Testbed and Settings

In our experimental testbed, we adopt 12 real datasets with diverse classes and sample sizes from UCI machine learning repository (http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/). These include 5 binary datasets (such as "ionosphere" and "sonar") and 7 multi-class datasets (such as "letter" and "poker"). We also include 6 well-known high-dimensional text datasets, including "Reuters-21578" and "Newsgroup" datasets, available at http://renatocorrea.googlepages.com/textcategorizationdatasets.

The details of our experimental testbed[3] are summarized in Table II.

We choose these datasets evaluating the performance of clustering algorithms for several considerations. Firstly, these datasets enjoy different properties. Some of them are binary-class, and some are multi-class. In addition, the feature dimensions are also quite diverse, ranging from a few dimensions (e.g. 6 for the "liver-disorders" dataset) to very high dimensional text data ($47,411$ for "newsgroup"). Further, these datasets are widely used as benchmark datasets for machine learning and data mining research. Finally, they are real data that allow us to examine the empirical performance of our technique in real applications.

[3]In the table, "index" is used to indicate the corresponding dataset to help subsequent discussions. D_a denotes datasets from the UCI datasets and D_b denotes datasets from the high-dimensional text datasets. The Reuter dataset was preprocessed by Renato Fernandes Corrêa, available at http://renatocorrea.googlepages.com/textcategorizationdatasets

index	dataset	#samples	#dimensions	#classes
D_{a1}	breast-cancer	683	10	2
D_{a2}	diabetes	768	8	2
D_{a3}	ionosphere	251	34	2
D_{a4}	liver-disorders	345	6	2
D_{a5}	sonar	208	60	2
D_{a6}	segment	2,310	19	7
D_{a7}	iris	150	4	3
D_{a8}	letter	15,000	16	26
D_{a9}	dna	2,000	180	3
D_{a10}	poker	25,010	10	10
D_{a11}	vehicle	846	18	4
D_{a12}	vowel	528	10	11
D_{b1}	w1a	47,272	300	2
D_{b2}	w2a	49,749	300	2
D_{b3}	w6a	17,188	300	2
D_{b4}	WebKB	4,291	19,687	6
D_{b5}	newsgroup	7,149	47,411	11
D_{b6}	Reuter	10,789	5,189	79

TABLE II

LIST OF DATA SETS IN OUR EXPERIMENTAL TESTBED. HERE "#" REPRESENTS "THE NUMBER OF".

In our experiments, we first randomly sample a subset of training data points (e.g. 10% of all data points), and then follow the previous work [40] to generate the side information by sampling the pairwise constraints from these data points according to their ground-truth class labels. Specifically, given the subset of randomly sampled data points, we generate all the positive pairs (every two training data points form a positive pair if they have the same class label), and randomly sample the same number of negative pairs from all possible negative pairs (every two training data points form a negative pair if they have different class labels).

For performing clustering, the number of clusters is simply set to the number of classes in the ground truth. The initial cluster centroids are randomly selected from the dataset. To enable fair comparisons, all comparing algorithms start with the same set of initial centroids. We repeat each clustering experiment for 20 times, and obtain the average results by averaging the results over these 20 runs.

B. Compared Methods

We compare the proposed two Bregman k-means algorithms (BKM-P2C and BKM-P2P) with other existing clustering techniques, including: a standard k-means with Euclidian distnace (Euc) as the baseline, the constrained k-means [35] with Euclidian distance (C-Euc) that only exploits pairwise constraints during online clustering, k-means with distance learned by RCA [3] (RCA), k-means with distance learned by DCA [17] (DCA), and k-means with distance learned by PGDM [40] (PGDM), the clustering method with information-theoretic metric learning (ITML) [8], and the k-means method with DistBoost metric learning (DistBoost) [34].

Table III shows a comparison of the competing methods in three aspects, where "index" represents each of the corresponding clustering methods to ease our subsequent discussions; "with side-info" indicates if the algorithm uses side information for clustering; "dimension" denotes if the algorithm is applicable to high dimensional data ("low" means it is only feasible for low-dimensional data, while "high" means it is applicable to both high and low dimensional data); "type-of-learning" indicates whether the algorithm is supervised or semi-supervised. The comparison shows that regular distance metric learning approaches are often feasible only for lower-dimensional data, while the proposed method can handle high dimensional data efficiently.

index	method	with side-info	dimension	type-of-learning
M_1	Euc	NO	high	unsupervised
M_2	C-Euc	YES	high	semi-supervised
M_3	RCA	YES	low	semi-supervised
M_4	DCA	YES	low	semi-supervised
M_5	PGDM	YES	low	semi-supervised
M_6	ITML	YES	low	semi-supervised
M_7	DistBoost	YES	low	semi-supervised
M_8	BKM-P2C	YES	high	semi-supervised
M_9	BKM-P2P	YES	high	semi-supervised

TABLE III

LIST OF COMPETING CLUSTERING ALGORITHMS.

C. Performance Metrics

To evaluate the clustering performance, we adopt some standard performance metrics, including pairwise Precision, pairwise Recall, and pairwise F1 measures [24]. The *pairwise precision, pairwise recall, and pairwise F1 measures* can be formally defined as follows:

$$\begin{aligned} Precision &= \frac{\#True\ Positive}{\#True\ Positive + \#False\ Positive} \\ Recall &= \frac{\#True\ Positive}{\#True\ Positive + \#False\ Negative} \\ F1 &= \frac{2 \times Precision \times Recall}{Precision + Recall} \end{aligned}$$

D. Experiment I: Evaluation with Fixed Number of Constraints

The first set of experiments is to evaluate the clustering performance given a fixed percent of randomly sampled points, (i.e., 10% in the experiments). First of all, different kinds of distance metrics are learnt based on the same set of constraints among these points, and then each metric is applied to the k-means clustering algorithm on each dataset.

Table IV and Table V show the comparison of average precision, recall, and F1 measurements of all the competing algorithms based on the same sets of constraints over different data sets. In the tables, "method" indicates different metric learning approaches used in the clustering algorithm. From the experimental results, we can see that, in general the proposed Bregman distance based k-means clustering approach outperforms other unsupervised clustering and semi-supervised clustering methods, and is significant on "diabetes", "sonar", "segment", "poker", "dna", and "vowel". In both tables, the top two highest F1 scores are highlighted in bold font. From the experimental results, we can observe that the proposed method achieves the best performance on most of the data sets, except for data set "liver-disorders", "iris", and "letter", where the ITML method obtains the best performance.

To further examine the significance of the performance gain, Table VIII shows the statistical t-test ($\alpha = 5\%$) evaluation between the proposed BKM and other methods on ordinary clustering. These tests show that in most data sets, BKM statistically outperforms the other methods significantly (5%). In particular, for 6 out of the 12 test sets, BKM significantly outperforms all other seven methods, while for the rest 6 data sets, BKM achieves the similar best performance that is comparable to the ITML method.

	breast			diabetes		
method	precision	recall	F1	precision	recall	F1
Euc	70.01±0.58	72.06±1.48	71.02±1.55	51.56±0.91	56.27±1.56	53.82±1.55
C-Euc	94.71±1.15	76.71±0.08	84.76±1.76	58.61±1.46	53.56±0.78	55.97±0.34
RCA	84.76±1.65	93.08±1.30	88.73±1.27	58.54±1.89	61.79±1.34	60.12±1.72
DCA	90.74±0.16	91.50±0.21	91.12±1.48	60.87±0.84	63.60±1.81	62.20±0.59
PDGM	89.07±0.90	81.07±0.82	84.88±0.68	56.10±0.21	55.33±0.88	55.71±0.43
ITML	95.87±1.50	85.86±1.49	90.59±1.47	73.82±0.44	67.76±0.21	70.66±1.96
DistBoost	92.74±1.95	92.95±0.78	92.84±1.86	62.25±0.84	68.31±1.38	65.14±0.40
BKM-P2C	98.20±1.00	94.53±1.47	96.33±0.08	99.23±1.87	64.65±0.79	78.29±1.30
BKM-P2P	98.78±0.82	98.09±0.41	98.43±0.31	96.42±1.28	63.24±0.13	76.38±0.69
	ionosphere			liver-disorders		
method	precision	recall	F1	precision	recall	F1
Euc	59.59±1.29	50.44±1.46	54.63±0.54	61.48±1.03	47.01±1.68	53.28±0.40
C-Euc	54.61±1.61	46.34±0.73	50.13±0.48	60.20±1.38	49.72±1.79	54.46±1.40
RCA	96.50±1.87	46.10±0.59	62.39±0.91	59.00±0.01	48.68±1.12	53.35±0.41
DCA	62.24±0.92	63.26±0.18	62.74±1.51	67.90±1.50	49.15±0.45	57.02±1.04
PDGM	61.10±0.84	63.56±0.38	62.30±0.33	92.37±0.98	48.94±1.36	63.98±0.47
ITML	96.49±0.73	58.00±0.11	72.45±1.51	91.09±0.65	51.67±1.22	65.94±0.71
DistBoost	73.85±1.87	64.25±1.36	68.72±1.46	49.01±0.69	49.35±0.80	49.18±0.93
BKM-P2C	95.83±1.36	58.86±0.37	72.93±1.55	94.59±1.54	49.76±1.29	65.22±1.64
BKM-P2P	97.24±1.96	62.55±1.22	76.13±0.63	95.41±1.55	50.40±1.92	65.96±1.72
	sonar			segment		
method	precision	recall	F1	precision	recall	F1
Euc	52.23±0.91	48.95±0.99	50.54±0.76	31.20±1.16	32.82±0.59	31.99±0.87
C-Euc	58.75±1.70	51.23±1.85	54.73±1.25	59.05±1.42	61.59±0.17	60.30±0.93
RCA	99.51±1.33	67.93±0.14	80.75±0.73	62.01±0.01	63.99±0.62	62.99±0.97
DCA	98.06±1.60	59.46±1.70	74.03±0.04	63.89±1.11	61.72±1.91	62.78±0.75
PDGM	95.98±0.37	69.62±1.11	80.70±0.41	61.31±1.38	59.78±0.58	60.54±0.38
ITML	97.65±1.73	55.50±0.77	70.78±0.97	62.35±1.00	60.29±1.04	61.31±1.28
DistBoost	75.90±1.81	74.04±0.53	74.96±1.23	66.18±1.17	61.88±1.40	63.96±0.41
BKM-P2C	98.76±0.65	74.21±0.64	84.75±0.20	65.59±0.50	69.54±1.43	67.51±0.57
BKM-P2P	98.98±1.21	72.91±1.13	83.96±1.45	70.20±0.37	67.13±0.04	68.63±0.53

TABLE IV

EVALUATION OF CLUSTERING PERFORMANCE (AVERAGE PRECISION, RECALL, AND F1) FOR NINE DIFFERENT CLUSTERING METHODS ON D_{a1}-D_{a6}. THE TOP TWO BEST F1 SCORES WERE HIGHLIGHTED WITH BOLD FONT IN EACH DATASET.

	iris			poker		
method	precision	recall	F1	precision	recall	F1
Euc	78.75±1.66	81.42±0.80	80.07±0.42	30.05±0.03	30.72±0.97	30.38±1.46
C-Euc	79.30±0.06	87.92±0.45	83.39±0.19	33.74±1.23	37.34±0.47	35.45±0.05
RCA	84.15±1.86	87.12±0.65	85.61±1.06	39.78±0.43	43.11±0.64	41.38±1.38
DCA	88.32±1.65	89.75±1.41	89.03±1.84	43.05±0.57	55.14±1.62	48.35±1.63
PDGM	77.67±1.60	87.56±1.29	82.32±1.11	56.40±0.63	42.58±0.41	48.52±0.63
ITML	98.38±0.69	95.56±0.75	96.95±0.11	50.33±1.38	48.88±1.72	49.60±0.52
DistBoost	89.56±0.32	87.70±1.66	88.62±1.49	47.24±0.56	47.53±1.94	47.39±1.49
BKM-P2C	96.39±1.52	89.98±0.40	93.07±1.40	57.48±1.42	62.12±1.66	59.71±0.32
BKM-P2P	98.30±0.88	93.12±0.52	95.64±1.23	57.25±1.20	61.09±0.08	59.11±1.84
	dna			letter		
method	precision	recall	F1	precision	recall	F1
Euc	60.86±0.23	64.54±0.62	62.65±0.31	11.70±0.63	20.77±0.79	14.97±0.61
C-Euc	69.39±1.56	68.25±1.99	68.81±1.46	21.79±1.11	16.81±1.82	18.98±1.82
RCA	69.60±1.80	72.21±1.03	70.88±0.54	25.55±0.83	22.80±0.90	24.09±1.76
DCA	72.91±0.39	68.83±0.89	70.81±1.94	23.66±1.28	25.59±1.73	24.59±1.41
PDGM	66.31±1.05	67.21±1.63	66.76±1.79	23.99±1.42	28.14±0.66	25.90±0.70
ITML	74.89±0.38	78.04±0.01	76.44±0.28	40.13±1.31	41.50±0.40	40.80±1.03
DistBoost	71.70±0.97	70.96±0.77	71.33±0.20	26.57±1.33	28.57±0.46	27.53±1.71
BKM-P2C	75.06±0.23	78.74±1.04	76.86±0.23	37.28±1.86	41.56±1.42	39.31±1.68
BKM-P2P	78.56±0.71	75.75±1.40	77.13±1.50	40.06±0.56	42.12±1.61	41.07±0.98
	vehicle			vowel		
method	precision	recall	F1	precision	recall	F1
Euc	29.08±1.56	31.34±0.38	30.17±0.31	24.75±0.36	24.67±1.83	24.71±0.66
C-Euc	54.17±0.11	52.95±0.68	53.56±0.62	50.86±0.56	53.98±1.18	52.38±1.90
RCA	39.17±1.58	43.96±1.82	41.43±1.67	35.18±1.92	30.25±1.64	32.53±0.25
DCA	30.43±0.21	33.66±0.77	31.96±0.02	22.41±1.66	27.84±1.49	24.83±0.57
PDGM	65.13±0.13	66.65±0.61	65.88±1.67	33.01±0.46	36.24±1.31	34.55±0.50
ITML	72.36±0.80	69.15±1.46	70.72±0.17	59.94±0.86	63.99±0.62	61.90±1.78
DistBoost	66.77±0.49	58.57±1.57	62.40±1.41	51.75±1.68	61.39±1.89	56.16±1.88
BKM-P2C	73.14±1.14	71.37±1.88	72.24±0.18	62.73±1.22	66.86±1.72	64.73±0.80
BKM-P2P	73.66±0.37	69.48±1.96	71.51±1.80	63.29±1.63	65.31±1.34	64.29±1.76

TABLE V

EVALUATION OF CLUSTERING PERFORMANCE (AVERAGE PRECISION, RECALL, AND F1) FOR NINE DIFFERENT CLUSTERING METHODS ON D_{a7}-D_{a12}.

E. Experiment II: Evaluation with Varied Number of Constraints

This experiment aims to study the influence of the pair-wise constraints for the text clustering performance. We randomly sample the positive and negative constraints among 10% of all points to 50% with increased 10% for each level. At each level, all the methods share the same constraints and random initializations. The average performances over 20 random initializations for each level are compared between the different methods to evaluate the influence of these constraints for each algorithm.

Fig. 2 shows the F1 scores of the compared algorithms on each of the constraint levels. Each sub-figure corresponds to one dataset. The horizontal axis is the number of constraints, and the vertical axis is the F1 score. This experiment is to compare the performance of all algorithms on various UCI datasets.

From Fig. 2, we observe that the number of constraints significantly affects the performance of data clustering. For most datasets, we observe an improvement in data clustering when the number of constraints is increased. We also observe that for most datasets, the proposed BKM method is able to gain more improvement in data clustering than the other methods, particularly when the number of constraints is large. This indicates that BKM is more effective in exploring the side information than the other methods. Finally, it is worthwhile pointing out that in some cases (i.e., "breast" dataset), the F1 score drops with the increasing number of constraint. The similar result was also found in some previous study [24].

F. Experiment III: Evaluation on High-dimensional Text Data

Due to the high computational complexity, some of the baseline methods are incapable of handling high dimensional data, such as texts. In this study, we restrict the comparison to the baseline methods that can deal with the high dimensional text datasets, i.e., RCA, DCA, and ITML. We also include the text clustering method based on cosine similarity (Cos for short).

Table VII summarizes the F1 scores on the "w1a", "w2a", "w6a", "WebKB", "20newsgroup", and "reuter21578" datasets. The best F1 scores are highlighted in bold font in Table VII. The results show that the proposed algorithm is applicable to learn distance functions for high dimensional data, and it outperforms the other baseline methods for semi-supervised clustering. Note that the baseline methods: RCA, DCA, ITML can only handle the first three datasets, and fail to learn appropriate distance metrics for the remaining high-dimensional text datasets due

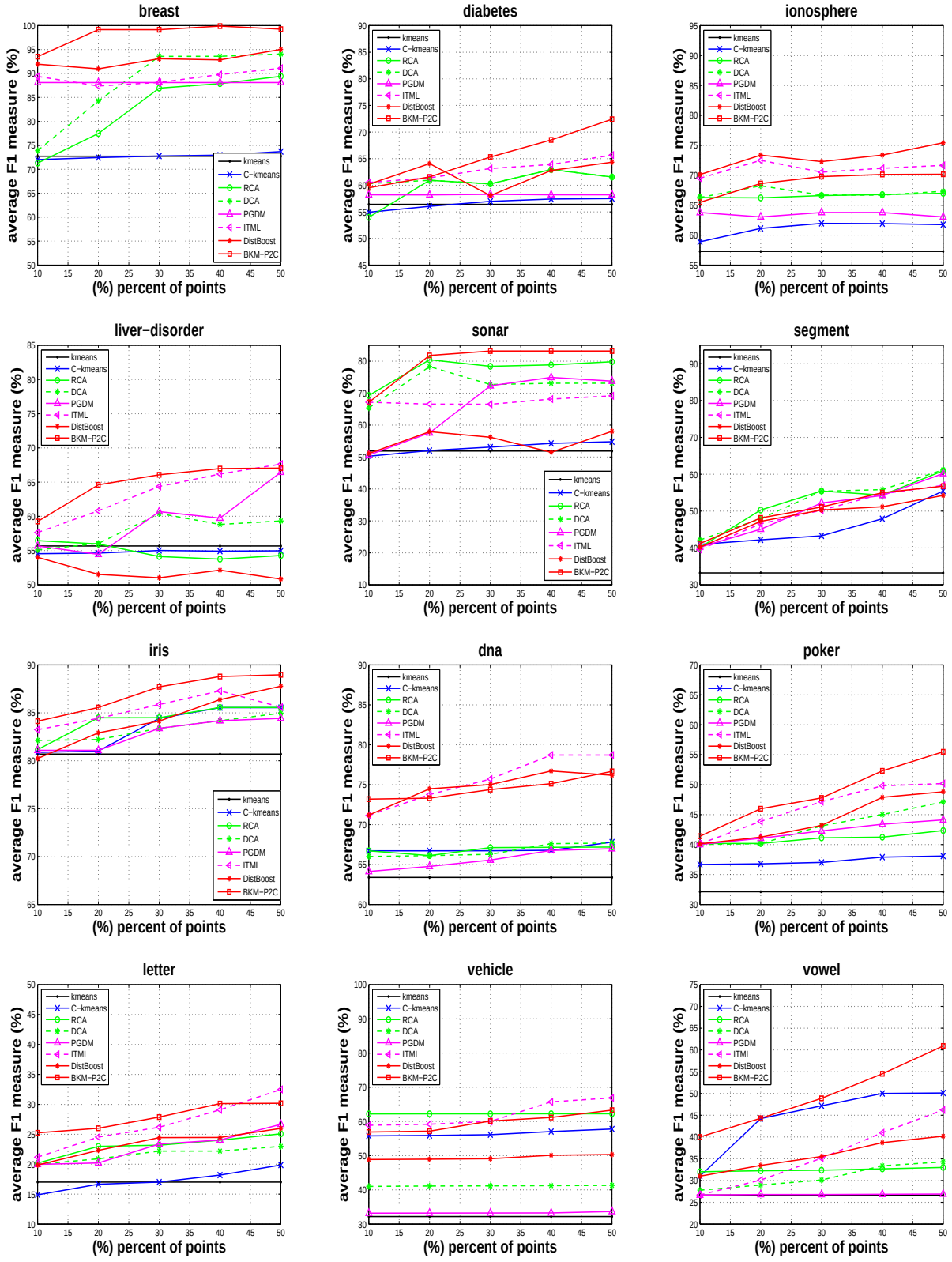


Fig. 2. Clustering performance comparison under different constraint levels (Continued). The x axis is the number of constraints, and the y axis is the average F1 score. The subfigures correspond to dataset $D_{a1} \sim D_{a12}$.

Clustering methods	w1a	w2a	w6a
Euc	71.26±0.29	69.36±0.82	72.13±0.41
Cos	72.28±0.11	69.58±0.62	74.35±0.55
C-Euc	80.55±0.53	85.55±0.92	69.93±0.99
C-Cos	81.18±1.01	87.73±0.89	69.85±1.06
RCA	83.34±1.07	88.59±1.05	84.49±1.91
DCA	84.29±1.01	88.17±1.18	85.51±0.97
ITML	86.15±0.38	89.91±1.07	86.67±1.91
BKM-P2C	87.87±0.55	92.98±0.78	94.49±0.41
BKM-P2P	**88.07**±0.56	**93.13**±0.86	**94.96**±0.29
Clustering methods	WebKB	newsgroup	reuter
Euc	32.38±0.36	14.15±0.28	40.71±0.41
Cos	32.71±0.93	15.81±0.18	42.13±0.03
C-Euc	57.76±0.96	15.51±0.38	48.87±0.88
C-Cos	59.27±0.87	16.13±0.34	47.58±0.69
RCA	OOM	OOM	OOT
DCA	OOM	OOM	OOT
ITML	OOT	OOM	OOT
BKM-P2C	69.91±0.72	21.91±0.31	59.91±1.07
BKM-P2P	**73.39**±0.55	**22.27**±0.49	**60.98**±1.02

TABLE VI

F1 PERFORMANCE OF K-MEANS CLUSTERING ON THE HIGH DIMENSIONAL TEXT DATA. "EUC" REPRESENTS THE APPLICATION OF THE EUCLIDIAN DISTANCE IN K-MEANS CLUSTERING. "COS" DENOTES THE COSINE DISTANCE IN K-MEANS CLUSTERING. "C-EUC" USES THE EUCLIDIAN DISTANCE IN CONSTRAINED K-MEANS, AND "C-COS" APPLIES THE COSINE DISTANCE IN CONSTRAINED K-MEANS. ONLY APPLICABLE METHODS ARE SHOWN. OOM INDICATES "OUT OF MEMORY", AND OOT INDICATES "OUT OF TIME". AVERAGE F1 SCORE IS USED IN THE EVALUATION.

to either the "out of memory" (OOM) or the "out of time" (OOT) errors. Among the available competing methods, BKM performs best on almost all datasets except the "w2a" data, where C-Euc achieves comparable performance as BKM.

There are several reasons to explain why Bregman distance is superior to Mahalanobis distance in both effectiveness and efficiency for high-dimensional data. Firstly, the Bregman distance

can be *nonlinear* by choosing some appropriate convex functions while conventional Mahalanobis distance is essentially a linear transformation of the original feature space that limits the capacity of forming flexible distance measures. Secondly, Bregman distance is locality sensitive to the sample pairs while Mahanalobis distance is in general stationary on the whole data set. Thirdly, when calculating the Bregman distance for any two samples x_a and x_b, we avoid the need of loading the whole metric/matrix A into memory, and require only for computing $(\nabla\varphi(x_a)-\nabla\varphi(x_b))$ in the memory according to Eq. (2). This is clearly more memory efficient, especially for higher dimensional data. Finally, for calculating the distance, traditional Mahanalobis distance $(x_a - x_b)^\top A(x_a - x_b)$ needs $O(d+1)$ vector multiplications; while the Bregman distance $(\nabla\varphi(x_a) - \nabla\varphi(x_b))^\top(x_a - x_b)$ only needs $O(1)$ vector multiplication which is thus computationally more efficient.

G. Experiment IV: Evaluation on Hierarchical Clustering

In this experiment, we evaluate the performance of distance learning methods with applications to hierarchical clustering. The *agglomerative* hierarchical clustering method is adopted in our study. In order to effectively evaluate the clustering result, we convert the hierarchical clustering result into a flat cluster by a threshold, which is automatically determined by the number of categories of the data. Both the UCI data sets and the text data sets are used in this experiment. Fig. 3 shows the comparison of F1 scores by varying the number of constraints from 100 to $1,000$. Table VII summarizes the F1 scores of hierarchical clustering using $1,000$ randomly selected constraints. Note "Euc" stands for the hierarchical clustering method in Euclidian space, and other methods are the application of the respective metric in hierarchical clustering.

According to the results in Fig. 3 and Table VII, we observe that the proposed approach overall outperforms the other baseline methods, for both UCI datasets and text datasets. We also examine the statistical significance. Table VIII shows the statistical t-test ($\alpha = 5\%$) results for hierarchical clustering. From the results, we found that the proposed method performs considerably better than the other methods on "breast", "liver", "sonar", "iris", and "poker" data sets. It achieves the similar performance as the ITML method on dataset "a1a", and "letter", and performs equally well as DCA on dataset "dna", "a1a", "ionosphere".

Finally, we also evaluate the performance on high dimensional text data as listed in Table VII. The encouraging results again verify the efficacy of the proposed method.

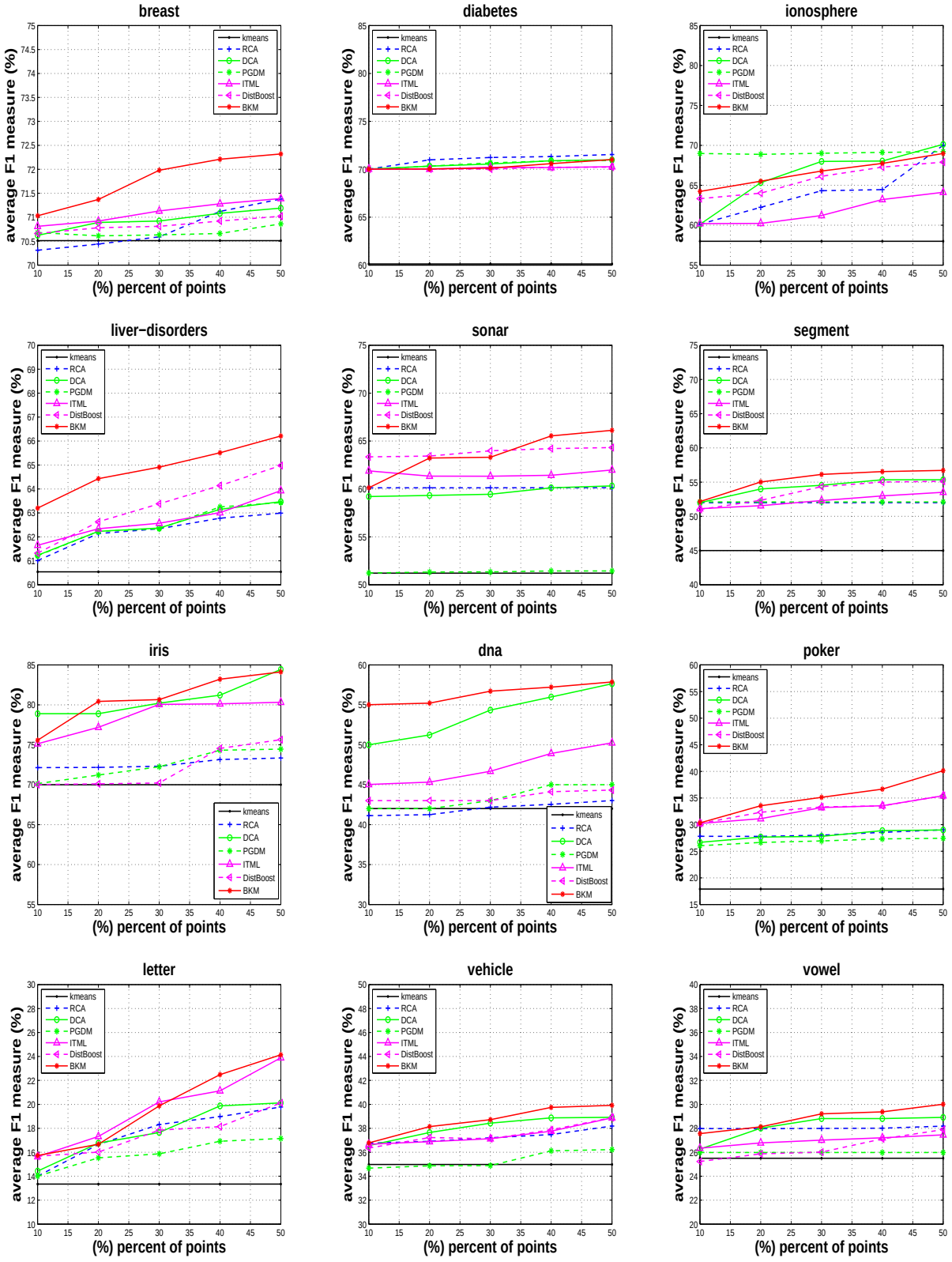


Fig. 3. Hierarchical clustering performance comparison under different constraint levels (Continued). The x axis is the number of constraints, and the y axis is the average F1 score. The subfigures correspond to dataset $D_{a1} \sim D_{a12}$.

Hierarchical clustering	w1a	w2a	w6a
Euc	89.27±1.08	89.92±1.87	74.83±1.19
Cos	90.12±1.11	91.05±1.21	89.92±1.03
RCA	93.08±1.02	93.82±1.12	91.98±0.99
DCA	92.19±1.09	93.38±0.86	93.32±1.02
ITML	92.92±0.94	94.33±1.28	93.32 ±0.86
Bregman	**94.54**±1.04	**96.76**±1.09	**96.96**±1.21
Hierarchical clustering	WebKB	newsgroup	reuter
Euc	78.22±1.45	22.85±1.92	61.34±1.17
Cos	80.54±1.12	23.95±1.22	62.56±1.31
RCA	OOM	OOM	OOT
DCA	OOM	OOM	OOT
ITML	OOT	OOM	OOT
Bregman	**81.91**±0.64	**27.79**±0.69	**66.96**±0.98

TABLE VII

F1 PERFORMANCE OF HIERARCHICAL CLUSTERING ON HIGH DIMENSIONAL TEXT DATA. OOM AND OOT INDICATES "OUT OF MEMORY" AND "OUT OF TIME", RESPECTIVELY. THE BEST RESULTS ARE MARKED IN **BOLD**.

H. Experiment V: Computational Complexity

The computational complexity for the clustering algorithm is mainly determined by both distance calculation and clustering scheme. For the k-means algorithm in the original Euclidian space, the distance between two points need $\mathcal{O}(d)$, where d is the dimension of the space. The total complexity for k-means is $\mathcal{O}(TKNd)$, where T is the number of iteration; K is the cluster number, and N is the point number. The computational complexity for the constrained k-means is the $\mathcal{O}(T(KN+n)d)$, where n is the number of pair-wise constraints.

The computational complexity of the Mahalanobis distance metric learning algorithms are determined by the training and clustering times. After projecting the points into the new RCA or DCA space, the clustering algorithm is almost the same with the original k-means algorithm. Thus the main computation lies on the training process. In the training process, both RCA and DCA will calculate the inverse of the covariance matrix. Besides, the DCA will also calculate the negative co-variant matrix. Calculating the inverse of large scale matrix is NP hard problem.

	t-test for k-means clustering results						
Dataset	M_1	M_2	M_3	M_4	M_5	M_6	M_7
D_{a1}	YES	YES	YES	YES	YES	YES	YES
D_{a2}	YES	YES	YES	YES	YES	YES	YES
D_{a3}	YES	YES	YES	YES	YES	NO	YES
D_{a4}	YES	YES	YES	YES	YES	NO	YES
D_{a5}	YES	YES	YES	YES	YES	YES	YES
D_{a6}	YES	NO	NO	NO	YES	YES	YES
D_{a7}	YES	YES	YES	YES	YES	NO	YES
D_{a8}	YES	YES	YES	YES	YES	NO	YES
D_{a9}	YES	YES	YES	YES	YES	YES	YES
D_{a10}	YES	YES	YES	YES	YES	NO	YES
D_{a11}	YES	YES	YES	YES	YES	YES	YES
D_{a12}	YES	YES	YES	YES	YES	NO	YES
	t-test for hierarchical clustering results						
Dataset	M_1	M_2	M_3	M_4	M_5	M_6	M_7
D_{a1}	YES	N/A	YES	YES	YES	YES	YES
D_{a2}	YES	N/A	NO	NO	NO	NO	NO
D_{a3}	YES	N/A	YES	NO	NO	YES	YES
D_{a4}	YES	N/A	YES	YES	YES	YES	YES
D_{a5}	YES	N/A	YES	YES	YES	YES	YES
D_{a6}	YES	N/A	YES	YES	YES	YES	NO
D_{a7}	YES	N/A	YES	NO	YES	YES	YES
D_{a8}	YES	N/A	YES	NO	YES	YES	YES
D_{a9}	YES	N/A	YES	YES	YES	YES	YES
D_{a10}	YES	N/A	YES	YES	YES	NO	YES
D_{a11}	YES	N/A	YES	YES	YES	YES	YES
D_{a12}	YES	N/A	YES	NO	YES	NO	YES

TABLE VIII

t-TEST RESULTS OF K-MEANS CLUSTERING AND HIERARCHICAL CLUSTERING RESULTS ($\alpha = 5\%$) BETWEEN BKM AND OTHER METHODS. M_i IS THE i-TH METHOD INDEXED IN TABLE III. D_j IS THE j-TH DATASET INDEXED IN TABLE II. "YES" AT METHOD M_i AND DATASET D_j REPRESENTS THE BREGMAN METHOD SIGNIFICANTLY OUTPERFORMS METHOD M_i ON DATASET D_j, AND "NO" REPRESENTS NON-SIGNIFICANT.

UCI	M_1	M_2	M_3	M_4	M_5	M_6	M_7
D_{a1}	< 0.01	0.02	0.59	0.10	72	0.10	0.09
D_{a2}	< 0.01	0.02	0.74	0.10	68	0.13	0.12
D_{a3}	< 0.01	0.01	0.46	0.02	8	0.18	0.14
D_{a4}	< 0.01	0.01	0.26	0.03	25	0.11	0.13
D_{a5}	< 0.01	0.01	0.52	0.03	27	0.09	0.10
D_{a6}	< 0.01	0.05	0.70	0.26	63	0.13	0.13
D_{a7}	< 0.01	< 0.01	0.09	0.01	8	0.03	0.04
D_{a8}	< 0.01	0.04	0.52	7.37	68	0.85	0.58
D_{a9}	< 0.01	0.36	1.37	0.04	33	0.54	0.67
D_{a10}	< 0.01	0.05	0.59	0.05	38	0.16	0.13
D_{a11}	< 0.01	0.02	0.41	0.11	65	0.11	0.10
D_{a12}	< 0.01	0.01	0.45	0.04	31	0.04	0.04
Average	**0.01**	**0.05**	**0.56**	**0.68**	**42**	**0.20**	**0.19**
Text	M_1	M_2	M_3	M_4	M_5	M_6	M_7
D_{b1}	1.22	3.41	n/a	2.88	n/a	3.13	3.24
D_{b2}	1.23	4.13	n/a	3.39	n/a	3.37	3.70
D_{b3}	0.54	1.32	n/a	1.18	n/a	0.16	0.14
D_{b4}	n/a	n/a	n/a	n/a	n/a	0.22	0.22
D_{b5}	n/a	n/a	n/a	n/a	n/a	12.99	11.39
D_{b6}	n/a	n/a	n/a	n/a	n/a	4.57	4.37
Average	**n/a**	**n/a**	**n/a**	**n/a**	**n/a**	**4.07**	**3.84**

TABLE IX

COMPARISON OF AVERAGE DISTANCE LEARNING TIME COST BY DIFFERENT METHODS (SECONDS).

So these methods are not applicable for large scale dataset.

For the Bregman distance function learning, we adopting the P2P scheme, we can calculate the P2P distance outside the iteration. Thus the computational complexity of the generating the P2P distance is $\mathcal{O}(N^2d)$. This computational complexity seems much larger than the P2C scheme, which only needs $\mathcal{O}(TKNd)$. However, once this distance matrix is generated, there is no need to calculate the distance between points or centers any more in the clustering iteration. So this process can be performed offline, and the online clustering process only needs

$\mathcal{O}(TK)$. Considering the online phase, the Bregman distance function with P2P scheme makes the clustering very efficient.

In this experiment, we evaluate the computational complexity of these algorithms by the overall running time, including both the time for training a distance function and the time for data clustering. The average time over the twelve UCI datasets of each algorithm is listed in Table IX, which shows that with the P2P scheme, the Bregman distance based clustering algorithm has a shorter running time than many baseline methods. We also list the average running time of the proposed algorithm for the six text datasets.

VI. Conclusions

In this paper, we proposed to learn Bregman distance functions for semi-supervised clustering algorithms using a non-parametric approach that is similar to SVM. Rather than learning some distance metric in the form of Mahalanobis distance, the Bregman distance function can handle high-dimensional data efficiently, and generalize squared Euclidean distance to a general metric space, which is a bijection between regular exponential families and regular Bregman divergences. We also incorporated the Bregman distance function into the k-means clustering algorithm and hierarchical clustering algorithms. Experiments of data clustering on a number of UCI datasets and several high dimensional text datasets have shown that the Bregman distance function outperforms other distance metric learning algorithms on both F1 measure and applicability. It also shows that increasing the side information for the learning process in general is able to boost the final clustering results. Finally, the statistical t-test on both k-means clustering and hierarchical clustering results showed that the proposed distance function outperforms the other regular distance metrics significantly in most cases.

Acknowledgements

The work was supported in part by Singapore MOE Academic Tier-1 Grant (RG67/07), National Science Foundation (IIS-0643494), and US Army Research Office (ARO Award W911NF-08-1-0403). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of MOE, and NSF.

REFERENCES

[1] K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Mach. Learn.*, 43(3):211–246, 2001.

[2] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with bregman divergences. In *Journal of Machine Learning Research*, pages 234–245, 2004.

[3] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

[4] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proc. 6th ACM SIGKDD Int. conference on Knowledge discovery and data mining*, pages 407–416, Boston, Massachusetts, US, 2000.

[5] L. Bregman. 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, 1967.

[6] Y. Chen and L. Tu. Density-based clustering for real-time stream data. In *KDD'07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, San Jose, California, USA, 2007.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.

[8] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML '07: Proceedings of the 24rd international conference on Machine learning*, pages 209–216, Corvalis, Oregon, 2007.

[9] P. Fraundorf. Thermal roots of correlation-based complexity. *Complex.*, 13(3):18–26, 2008.

[10] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Elsevier, 1990.

[11] G. Giannopoulos, T. Dalamagas, M. Eirinaki, and T. Sellis. Boosting the ranking function learning process using clustering. In *Proc. 10th ACM workshop on Web information and data management*, pages 125–132, Napa Valley, CA, 2008.

[12] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems*, 2005.

[13] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[14] X. He, D. Cai, S. Yan, and H.-J. Zhang. Neighborhood preserving embedding. In *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 2, pages 1208–1213, 2005.

[15] S. C. Hoi, W. Liu, and S.-F. Chang. Semi-supervised distance metric learning for collaborative image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008)*, June 2008.

[16] S. C. Hoi and M. R. Lyu. A multimodal and multilevel ranking scheme for large-scale video retrieval. *IEEE Transactions on Multimedia (TMM)*, 10(4):607–619, 2008.

[17] S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[18] S. C. H. Hoi, M. R. Lyu, and R. Jin. A unified log-based relevance feedback scheme for image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):509–204, 2006.

[19] G. H. J. Goldberger, S. Roweis and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, 2005.

[20] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[21] H. Jeffreys and B. S. Jeffreys. Mean-value theorems. *1.13 in Methods of Mathematical Physics,*, pages 49–50, 1988.

[22] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[23] J. Lee and C. Zhang. Classification of gene-expression data: The manifold-based metric learning way. *Pattern Recogn.*, 39(12):2450–2463, 2006.

[24] Y. Liu, R. Jin, and A. K. Jain. Boostcluster: boosting clustering by pairwise constraints. In *The 13th ACM SIGKDD Conference On Knowledge Discovery and Data Mining*, pages 450–459, San Jose, California, USA, 2007.

[25] A. Lukasova. Hierarchical agglomerative clustering procedure. *Pattern Recognition (PR)*, 11(5–6):365–381, 1979.

[26] D. Maesschalck, R. D. Jouan-Rimbaud, and D. Massart. The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50:1–18, 2000.

[27] P. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2:49–55, 1936.

[28] J. K. I. Y. K. Z. O. Perera, D.; Kay. Clustering and sequential pattern mining of online collaborative learning data. *Knowledge and Data Engineering, IEEE Transactions on*, Issue 99:1 – 1, 2008.

[29] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[30] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 807–814, Corvalis, Oregon, 2007.

[31] L. Si, R. Jin, S. C. H. Hoi, and M. R. Lyu. Collaborative image retrieval via regularized metric learning. *ACM Multimedia Systems Journal*, 12(1):34–44, 2006.

[32] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[33] J. B. Tenenbaum and V. de Silva andJohn C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[34] T. H. Tomboy, A. Bar-hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *ICML '04: Proceedings of the 21st International Conference on Machine Learning*, pages 393–400, 2004.

[35] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *ICML '01: Proceedings of the 18rd international conference on Machine learning*, pages 577–584, 2001.

[36] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 1473–1480, 2006.

[37] K. Weinberger and G. Tesauro. Metric learning for kernel regression. In *Eleventh International Conference on Artificial Intelligence and Statistics*, pages 608–615, Omnipress, Puerto Rico.

[38] L. Wu, X.-S. Hua, N. Yu, W.-Y. Ma, and S. Li. Flickr distance. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 31–40, Vancouver, British Columbia, Canada, 2008.

[39] L. Wu, L. Yang, N. Yu, and X.-S. Hua. Learning to tag. In *18th International World Wide Web Conference*, pages 361–361, April 2009.

[40] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, 2002.

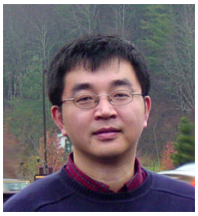
[41] L. Yang, R. Jin, R. Sukthankar, and Y. Liu. An efficient algorithm for local distance metric learning. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI)*, 2006.

[42] K. Yi, F. Li, G. Kollios, and D. Srivastava. Efficient processing of top-k queries in uncertain databases with x-relations. *IEEE Trans. on Knowl. and Data Eng.*, 20(12):1669–1682, 2008.

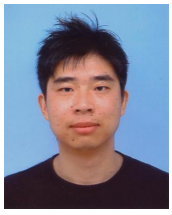
[43] D.-C. Zhan, M. Li, Y.-F. Li, and Z.-H. Zhou. Learning instance specific distances using metric propagation. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1–8, Montreal, Quebec, Canada, 2009.

Lei Wu received the Bachelor degree in Special Class for Gifted Young (SCGY) from University of Science and Technology of China (USTC), from which he is now pursuing his Ph.D degree in Electronic Engineering and Information Science. From 2006 to 2008, he had been a research intern at Microsoft Research Asia working on image annotation and tagging. From 2008 to 2009, he was visiting Nanyang Technological University working with Dr. Steven C.H. Hoi. His research interests include machine learning, multimedia retrieval, and computer vision. He received Microsoft Fellowship in 2007.

Steven C.H. Hoi is currently an Assistant Professor in the School of Computer Engineering, Nanyang Technological University, Singapore. He received his Bachelor degree in Computer Science from Tsinghua University, Beijing, P.R. China, and his Master and Ph.D degrees in Computer Science and Engineering from Chinese University of Hong Kong. His research interests include statistical machine learning, multimedia information retrieval, web search and data mining. He is a member of IEEE and ACM.



Rong Jin is currently an Associate Professor in the department of Computer Science and Engineering at Michigan State University. He received his Ph.D. degree in Computer Science from Carnegie Mellon University, 2003. His research interests are statistical learning and its application to large-scale information management, including web text retrieval, content-based image retrieval, gene regulatory network reconstruction, neuron data analysis, and visual object recognition.



Jianke Zhu obtained Bachelor degree in Mechatronics and Computer Engineering from Beijing University of Chemical Technology in 2001. He received his Master degree in Electrical and Electronics Engineering from University of Macau in 2005. He received his PhD degree in the Computer Science and Engineering department at the Chinese University of Hong Kong. His research interests are in pattern recognition, computer vision and statistical machine learning.

Nenghai Yu is currently a Professor in the Department of Electronic Engineering and Information Science of University of Science and Technology of China (USTC). He is the Executive Director of MOE-Microsoft Key Laboratory of Multimedia Computing and Communication, and the Director of Information Processing Center at USTC. He graduated from Tsinghua University,Beijing, China, and obtained his M.Sc. Degree in Electronic Engineering in 1992, and then he joined in USTC and worked there until now. He received his Ph.D. Degree in Information and Communications Engineering from USTC, Hefei, China, in 2004. His research interests are in the field of multimedia information retrieval, digital media analysis and representation, media authentication, video surveillance and communications etc.