Published in final edited form as:
IEEE Trans Pattern Anal Mach Intell. 2012 December ; 34(12): 2407–2419. doi:10.1109/TPAMI.2012.44.

Shape Retrieval Using Hierarchical Total Bregman Soft Clustering

Meizhu Liu [Member, IEEE],
Dept. of CISE, University of Florida, USA

Baba C. Vemuri [Fellow, IEEE],
Dept. of CISE, University of Florida, USA

Shun-ichi Amari [Fellow, IEEE], and
Riken Brain Science Institute, Japan

Frank Nielsen [Senior Member, IEEE]
Ecole Polytechnique, Paris, FR and Sony Computer Science Laboratories, Tokyo, JP

Abstract

In this paper, we consider the family of total Bregman divergences (tBDs) as an *efficient* and *robust* "distance" measure to quantify the dissimilarity between shapes. We use the tBD based ℓ_1-norm center as the representative of a set of shapes, and call it the *t*-center. First, we briefly present and analyze the properties of the tBDs and *t*-centers following our previous work in [1]. Then, we prove that for any tBD, there exists a distribution which belongs to the *lifted* exponential family of statistical distributions. Further, we show that finding the maximum a posteriori estimate of the parameters of the lifted exponential family distribution is equivalent to minimizing the tBD to find the *t*-centers. This leads to a new clustering technique namely, the total Bregman soft clustering algorithm. We evaluate the tBD, *t*-center and the soft clustering algorithm on shape retrieval applications. Our shape retrieval framework is composed of three steps: (1) extraction of the shape boundary points (2) affine alignment of the shapes and use of a Gaussian mixture model (GMM) [2], [3], [4] to represent the aligned boundaries, and (3) comparison of the GMMs using tBD to find the best matches given a query shape. To further speed up the shape retrieval algorithm, we perform hierarchical clustering of the shapes using our total Bregman soft clustering algorithm. This enables us to compare the query with a small subset of shapes which are chosen to be the cluster *t*-centers. We evaluate our method on various public domain 2D and 3D databases, and demonstrate comparable or better results than state-of-the-art retrieval techniques.

Index Terms

total Bregman divergence; *t*-center; lifted exponential families; hard clustering; soft clustering; Gaussian mixture model; 3D shape retrieval

1 Introduction

As the number of images on the Internet, in public databases and in biometric systems grows larger and larger, efficient and accurate search algorithms for retrieval of the best matches have become crucial for a variety of tasks. Therefore, image retrieval becomes more and more fundamental in computer vision and plays an indispensable role in many potential applications. In contemporary literature, there are mainly two types of algorithms for image

retrieval, key-words based and content based. Key-words are an important and easy to use features for representation and retrieval of images. However, though efficient, key-words are very subjective, since different people may use different key-words to index the same image. Therefore, the accuracy of key-words based retrieval is very limited. Hence there is interest in the idea of retrieval based on image features [5], [6], [7], [8], [9] such as texture, color, shape, and so on. Of these, shape is considered more generic and is one of the best for recognition as studies [10] have shown. Shape comparison and classification is very often used in the areas of object detection [11], [12] and action recognition [13]. Therefore many researchers [14], [15], [16], [17], [18], [19], [20], [21], [22] have been developing algorithms for improving the performance of shape retrieval. An efficient modern shape retrieval scheme has the following two components: an accessible and accurate shape representation, and an efficient as well as robust distance/divergence measure. There are many ways to represent shapes, for example, axial representation [23], [24], primitive-based representation [25], constructive representation [26], reference points and projection based representation [27], cover-based representation [28], histograms of oriented gradients [29] etc. Of these, contour based representation in object recognition methods [30], [31], [32], [33], [34], [35], [36] have shown great performance. Probability density function (pdf) has emerged as a widely used and successful representation for shape contours [1], [5], [37]. It is known to be mathematically convenient and robust to rigid transformations, noise, occlusions and missing data. Bearing this in mind and following our previous work in [1], we choose to represent shapes by pdfs in this paper.

The large size of image/shape databases today need faster retrieval algorithms (e.g., TinEye reverse image search and Google image retrieval, both require real-time response). In this paper, we present a fast and accurate shape retrieval method, which represents shapes using Gaussian mixture models (GMMs) [2], [3], [4]. The shapes are divided into smaller groups using a total Bregman divergence soft clustering algorithm, where each cluster is represented by a tBD based ℓ_1-norm center, called the t-center [1], [38]. As shown in [1], [38], the t-center is a weighted combination of all elements in the cluster. Moreover, it has a closed form expression, and is robust to noise and outliers. For the readers' convenience, we will revisit all of these points in Section 3.

This paper is a significant extension of our conference paper on shape retrieval using tBD [1]. tBD is derived from total least squares [39] that has been used in linear regression to fit a line/plane to a set of points. Total least squares based linear regression seeks to minimize the orthogonal distance from every point to the line/plane, whereas ordinary least squares linear regression seeks to minimize the ordinate distance from each point to the line/plane. Fig. 1(a) and (b) show the difference between total least squares and least squares before and after transforming the coordinate system. In Fig. 1, the dotted green lines represent least squares and the solid red lines correspond to total least squares. We can see that when the coordinate system is rotated, least squares solution will change, but total least squares will remain the same. Therefore, it easy to see (and prove) that least squares regression is coordinate-system dependent. One of the motivations for defining and using the tBD is to remove this dependence on the coordinate system.

In [38], we defined tBD along with its induced ℓ_1-norm based t-center, and derived some of its properties, e.g. statistical robustness to noise and outliers, and a closed-form expression for the t-center. We introduced the tBD hard clustering algorithm in [1], and used this algorithm to divide a large database into hierarchical clusters, where the objects in the same cluster are more similar than objects from different clusters. In this paper, we will introduce other tBD-based centers, including the arithmetic center, and its special cases: the geometric center and the harmonic center. Further, we show that to every tBD, there corresponds a lifted exponential distribution and prove that estimating the parameters of this distribution

using the MAP estimator is equivalent to finding the t-center by minimizing the tBD. Based on this theoretical result, we present the tBD *soft* clustering algorithm for use in shape retrieval.

We evaluate the soft clustering algorithm on synthetic data sets and real image databases, and compare it to total Bregman hard clustering, as well as Bregman hard and soft clustering algorithms. Additionally, we apply tBD soft clustering to the task of shape retrieval applied to several shape databases. For this, we divide the database into smaller clusters and use the t-center to represent each cluster. The only thing that needs to be stored are the t-centers, which is achieved using a k-tree, (a hybrid of the B-trees and k-means) [40], [41]. During retrieval, we only need to compare the query with the t-centers, and prune the clusters whose t-centers have large dissimilarity with the query. This process significantly reduces the number of unnecessary comparisons, and greatly speeds up the retrieval. We show that this method also performs well on various 2D shape databases such as the MPEG-7 database [42], the Brown database [14] and the Swedish leaf data set [43], as well as on 3D shape databases such as the Princeton Shape Benchmark (PSB) database [44].

The rest of this paper is organized as follows. In Section 2 we review the conventional Bregman divergence and the newly proposed tBD along with their properties. Section 3 briefly introduces different types of tBD centers, the ℓ_p-norm arithmetic center along with its special cases (geometric center, harmonic center). We will mainly focus on the t-center, which is the ℓ_1-norm based tBD median, and delve into its properties. In section 4, we present the key theoretical result namely, we show that for every tBD, there corresponds a lifted exponential distribution, and using the maximum a posteriori (MAP) estimation to estimate the parameters of this distribution corresponds to minimizing the tBDs and finding the t-centers. This naturally leads to the tBD soft clustering algorithm, which is presented in Section 5. Section 6 describes the detailed shape retrieval experimental design, and the application of tBD and the t-center on the classical 2D shape databases, including the MPEG-7, Brown and Swedish leaf databases, and also on the 3D PSB database. We quantitatively compare our results with those from other techniques. Finally, we draw conclusions in Section 7.

2 Total Bregman Divergences

In this section, we first recall the definitions of BD [45] and tBD [38]. Both divergences are dependent on a convex and differentiable function $f : X \to \mathbb{R}$ that generates the divergences. It is worth pointing out that if f is not differentiable, one can mimic the definition and proofs of properties with gradient substituted by any of its subdifferentials [46], and the gradient can be any value contained inside the interval of the subderivatives. Using this subderivative will retain the properties of tBD.

Definition 2.1—[45] The Bregman divergence d associated with a real valued strictly convex and differentiable function f defined on a convex set X between points $x, y \in X$ is given by,

$$d_f(x, y) = f(x) - f(y) - \langle x - y, \nabla f(y) \rangle, \quad (1)$$

where $\nabla f(y)$ is the gradient of f at y and $\langle \cdot, \cdot \rangle$ is the inner product determined by the space on which the inner product is being taken.

$d_f(\cdot, y)$ measures the error using the tangent function at y to approximate f, and can be seen as the distance between the first order Taylor approximation to f at y and the function

evaluated at x. As shown in Fig. 2, Bregman divergence measures the ordinate distance, the length of the dotted green line which is parallel to the y-axis. It is dependent on the coordinate system, for example, if we rotate the coordinate system, the ordinate distance will change (see the dotted lines in Fig. 2(a) and (b)). This coordinate dependent distance has great limitations because it requires a fixed coordinate system. This is unrealistic in the cases where a fixed coordinate system is difficult to build. With the motivation to overcome this shortcoming and free ourselves from choosing coordinate systems, we proposed total Bregman divergence.

Definition 2.2—[1] The total Bregman divergence δ associated with a real valued strictly convex and differentiable function f defined on a convex set X between points $x, y \in X$ is defined as,

$$\delta_f(x,y)=\frac{f(x)-f(y)-\langle x-y,\nabla f(y)\rangle}{\sqrt{1+\left\|\nabla f(y)\right\|^2}}, \quad (2)$$

$\langle \cdot, \cdot \rangle$ is inner product as in Definition 2.1, and $\|\nabla f(y)\|^2 = \langle \nabla f(y), \nabla f(y) \rangle$ generally.

As shown in Fig. 2, tBD measures the orthogonal distance, and if we translate or rotate the coordinate system, $\delta(\cdot, \cdot)$ will not change.

Compared to the BD, tBD contains a weight factor (the denominator) which complicates the computations. However, this structure brings up many new and interesting properties and makes tBD an "adaptive" divergence measure in many applications. Note that, in practice, X can be an interval, the Euclidean space, the Riemannian space, functions [47]. Also tBD is not symmetric i.e., $\delta_f(x, y) \neq \delta_f(y, x)$, but we can make it symmetric very easily in many ways, e.g. total Jensen-Bregman divergence [48]

$$\delta_f s=(\delta_f(x,(x+y)/2)+\delta_f(y,(x+y)/2))/2 \quad (3)$$

or

$$\delta_f s=(\delta_f(x,y)+\delta_f(y,x))/2 \quad (4)$$

or

$$\delta_f s=\sqrt{\delta_f(x,y)\times\delta_f(y,x)}. \quad (5)$$

Table 1 lists some tBDs with various associated convex functions.

3 Total Bregman divergence based centers

In many applications of computer vision and machine learning such as image and shape retrieval, clustering and classification etc., it is common to seek a representative or template for a set of objects having similar features. This representative normally is a cluster center, thus, it is desirable to seek a center that is intrinsically representative and easy to compute. In this section, we will introduce the tBD-based centers, including the ℓ_p-norm mean, the geometric and harmonic means respectively. Specifically, we will focus on the ℓ_1-norm cluster center that we call the total center (*t-center* for short) and explore its properties.

Some of these centers were introduced in [1], [38] but are included here to make this paper more self contained.

Definition 3.1—Let $f: X \to \mathbb{R}$ be a convex and differentiable function and $E = \{x_1, x_2, \cdots, x_n\}$ be a set of n points in X, then, the ℓ_p-norm distance based on tBD, $\mathscr{A}_f^p$, between a point $x \in X$ and E with associated f and the ℓ_p-norm is defined as

$$\mathscr{A}_f^p(x, E) = \left(\frac{1}{n} \sum_{i=1}^{n} (\delta_f(x, x_i))^p \right)^{1/p} \quad (6)$$

The ℓ_p-norm mean $\bar{x}_p$ of E is defined as

$$\bar{x}_a^p = \arg\min_x \mathscr{A}_f^p(x, E). \quad (7)$$

It is well known that the conventional geometric, harmonic and arithmetic means (in the Euclidean case) have a strong relationship. This is also the case for the tBD centers. When $p = 1$, the tBD center is the arithmetic mean of δ_f, and when $p = -1$, the tBD mean becomes the harmonic mean, and when $p \to 0$, the mean becomes the geometric mean [49].

These means also bear the name of circumcenter ($p \to \infty$), centroid ($p = 2$) and median ($p = 1$) respectively [50]. In this paper, we call the median ($p = 1$) the t-center and we will present an analytic form for the t-center and focus on its applications to shape retrieval. We would like to mention that the ℓ_2-norm center for tBD is not in closed form and is not as robust as ℓ_1-norm counterpart, thus motivating us to seek the ℓ_1-norm t-center.

3.1 ℓ_1-norm t-center

Given a set $E = \{x_1, x_2, \cdots, x_n\}$, we can obtain the ℓ_1-norm t-center $\bar{x}$ of E by solving the following minimization problem

$$\bar{x} = \arg\min_x \delta_f^1(x, E) = \arg\min_x \sum_{i=1}^{n} \delta_f(x, x_i) \quad (8)$$

Using the ℓ_1-norm t-center $\bar{x}$ has advantages over other centers since it has a closed form which makes it computationally attractive. The advantage is evident in the experiments presented subsequently.

The t-center is closely related to other kinds of tBD-based centers, like the geometric mean and harmonic mean. We will show in the next section that, based on tKL, the t-center of a set of pdfs is a weighted geometric mean of all pdfs, and the t-center of a set of symmetric positive definite matrices is the weighted harmonic mean of all matrices.

3.2 Properties of t-center

In [38], we proved that the t-center exists, is unique and can be written in an explicit form. The proof made use of the convexity of f and the the Legendre dual space of tBD. We will now illustrate this result using examples. But prior to doing that, we present some definitions which are used in the example illustrations.

Definition 3.2—Let $x \in X$, (X can be $\mathbb{R}^n$, $\mathbb{R}_+^n$, or the set of probability distributions in $\mathbb{R}_+^n$, i.e. $\sum_{i=1}^{n} x_i=1, x_i>0$) and $f(x)$ be a convex function. We then have the dual coordinates through the Legendre transformation

$$x^*=\nabla f(x), \quad (9)$$

and the dual convex function

$$f^*(x^*)=sup_x\{\langle x^*, x\rangle - f(x)\}. \quad (10)$$

For the Legendre transformation, the derivative of the function f becomes the argument to the function f^*. In addition, if f is convex, then f^* satisfies the functional equation

$$f^*(\nabla f(x))=\langle x, \nabla f(x)\rangle - f(x). \quad (11)$$

The Legendre transformation is its own inverse, i.e. $f^{**} = f$. If f is a closed (lower-continuous) convex function, then f^* is also closed and convex.

We already know that the gradient at the ℓ_1-norm t-center $\bar{x}$ is a weighted Euclidean average of the gradient of all the elements in the set $E = \{x_1, x_2, \cdots, x_n\}$ [38], as in

$$\nabla f(\bar{x})=\left(\sum_{i=1}^{n} w_i \nabla f(x_i)\right) / \left(\sum_{i=1}^{n} w_i\right), \quad (12)$$

with the weights $w_i = (1 + \|\nabla f(x_i)\|^2)^{-1/2}$. Utilizing the Legendre dual transformation, and let f^* be the Legendre dual function of f, i.e.

$$f^*(y)=\sup_x\{yx - f(x)\}, \quad (13)$$

then

$$\bar{x}=\nabla f^*(y_0), \quad (14)$$

and

$$y_0=\left(\sum_{i=1}^{n} w_i \nabla f(x_i)\right) / \left(\sum_{i=1}^{n} w_i\right), \quad (15)$$

which is the weighted average of gradients. For example, if $f(x) = x^2$, then

$$f^*(y)=\sup_x\{\langle y, x\rangle - f(x)\}=\frac{y^2}{2}, \quad (16)$$

and

$$\bar{x}=y_0, \quad (17)$$

where

$$y_0=\left(2\sum_{i=1}^{n} w_i x_i\right) / \left(\sum_{i=1}^{n} w_i\right), \quad (18)$$

and

$$w_i=\frac{1}{\sqrt{1+4\left\|x_i\right\|^2}}. \quad (19)$$

The t-center has a closed form expression, which is a weighted average, and the weight is inversely proportional to the magnitude of the gradient of f at the corresponding element. Table 1 lists the t-centers corresponding to various associated (commonly used) convex functions. Obviously, this list is not exhaustive and there are convex functions for which the t-centers are not in closed form and for certain classes of convex functions, the computation of the inverse can be intractable [51]. In fact computation of the inverse in this context is the focus of many machine learning algorithms [52], [53]. Table 1 also contains a column (# 5) showing the ℓ_1-norm centers derived using the BD [45] corresponding to the given convex functions in column 2. Note that regardless of the chosen convex generating function, BD center is the same for all of them. This is in sharp contrast to the t-centers, which we believe are a richer class.

Also, as an ℓ_1-norm mean, the t-center is closely related to the geometric mean and the harmonic mean. The relationship is obvious when using the tKL between two pdfs. Let $f(q) = \int q(x) \log q(x) dx$, which is the negative entropy [54], and $E = \{q_1, q_2, \cdots, q_n\}$ be a set of pdfs, the t-center is then given by

$$\bar{q}=c\prod_{i}^{n} q_i^{w_i/\sum_j w_j},\ w_i=\frac{1}{\sqrt{1+\int(1+\log q_i(x))^2 q_i(x)dx}}, \quad (20)$$

where c is a normalization constant to make $\bar{q}$ a pdf, i.e. $\int \bar{q}(x)dx = 1$. $\bar{q}$ is a weighted geometric mean of $\{q_i\}_{i=1}^n$ which belongs to the exponential family generated by $q_1, \cdots, q_n$. This is very useful in order two tensor interpolation, where the order two tensor is a symmetric positive definite (SPD) matrix. The tBD between two such tensors Q_i and Q_j (which can be considered as the covariances of two distinct normal distributions) can be taken as the tKL between two normal distributions

$$p(\mathbf{x};Q_i)=\frac{1}{\sqrt{(2\pi)^d \det Q_i}}\exp\left(-\frac{1}{2}\mathbf{x}' Q_i^{-1}\mathbf{x}\right), \quad (21)$$

$$q(\mathbf{x};Q_j)=\frac{1}{\sqrt{(2\pi)^d \det Q_j}}\exp\left(-\frac{1}{2}\mathbf{x}' Q_j^{-1}\mathbf{x}\right), \quad (22)$$

and

$$tKL(Q_i, Q_j) = tKL(p, q) = \frac{\log(\det(Q_i^{-1}Q_j)) + tr(Q_j^{-1}Q_i) - d}{2\sqrt{c + \frac{(\log(\det Q_j))^2}{4} - \frac{d(1+\log 2\pi)}{2}\log(\det Q_j)}},$$

where $c = \frac{3d}{4} + \frac{d^2 \log 2\pi}{2} + \frac{(d \log 2\pi)^2}{4}$, and d is the number of rows/columns of Q_i. The t-center $\bar{Q}$ for $\{Q_i\}_{i=1}^n$ is the weighted harmonic mean:

$$\bar{Q} = \left(\sum_{i=1}^{n} \frac{w_i^{-1}}{Q_i}\right)^{-1}, \quad w_i = \frac{\left(2\sqrt{c + \frac{(\log(\det Q_i))^2}{4} - \frac{d(1+\log 2\pi)}{2}\log(\det Q_i)}\right)^{-1}}{\sum_j \left(2\sqrt{c + \frac{(\log(\det Q_j))^2}{4} - \frac{d(1+\log 2\pi)}{2}\log(\det Q_j)}\right)^{-1}}. \quad (23)$$

Besides the closed form expression, another fundamental property of the t-center is that it is provably robust to outliers (see [38] for the analysis of the influence function). We will state its theoretical robustness here in the form of a theorem that was proved in [38] and depict its robustness property in practice through examples in the experimental section.

Theorem 3.1—[38] The t-center is statistically robust to outliers. The influence function of the t-center from outliers is upper bounded.

Using the ℓ_1-norm t-center $\bar{x}$ has advantages over other centers resulting from the norms with $p > 1$ in the sense that, besides robustness, $\bar{x}$ has an analytic form which makes it computationally attractive. This advantage is explicitly evident in the applications of clustering described later.

4 tBDs and lifted exponential families

In this section, we will elucidate a stochastic aspect of tBD by proving a relationship between tBD and lifted exponential family distributions. The l_1-norm t-center corresponds to the Bayesian Maximum a posteriori estimation (MAP) estimator of the associated probability distribution.

In probability theory and statistics, a family of probability density functions or probability mass functions is said to be an exponential family (EF) if it can be expressed in the following standard form,

$$p(\mathbf{x};\theta) = h(\mathbf{x})\exp\left(\langle\theta, \mathbf{x}\rangle - g(\theta)\right) \quad (24)$$

where θ is called the natural parameter, which is a vector lying in the natural parameter space, $h(\mathbf{x}) > 0$ is a dominating measure, $\mathbf{x}$ is a vector-valued random variable, $\langle\theta, x\rangle$ is the inner product $\Sigma_i \theta_i x_i$ and $g(\theta)$ is a convex function of θ. $g(\theta)$ is called the log partition function or cumulant generating function. There are two common types of exponential families [55], including the continuous families (e.g. normal, Gamma, Beta, Log-normal, Weibull and inverse Gaussian distributions), and discrete families (e.g. Poisson, Binomial,

and Multinomial distributions). In the case of discrete families, $h(\mathbf{x})$ is a counting measure and $\int$ will be replaced by Σ.

A multivariate parametric family of distributions $p(\mathbf{x}, \theta)$ in (24) is said to be a regular exponential family, when $\mathbf{x}$ is a minimal sufficient statistic [56], [57] and $g(\theta)$ is convex and strictly differentiable. In this paper, we will use the regular exponential family distributions.

Banerjee et al. [45] have shown that there is a bijection between the regular exponential family distributions and BDs such that

$$p(\mathbf{x};\theta)=b(\mathbf{x})\exp\left(-d_f(\mathbf{x},\eta(\theta))\right), \quad (25)$$

where f is the Legendre conjugate function of g, $\eta = \nabla g(\theta)$, and $b(\mathbf{x}) = h(\mathbf{x}) \exp(-f(\mathbf{x}))$.

There is a bijection between BD and tBD such that every BD corresponds to a unique convex differentiable function f and every convex differentiable f corresponds to a unique tBD and vice versa. According to the bijection transitivity, we can recover the exponential family distribution (25) as

$$p(\mathbf{x};\theta)=b(\mathbf{x})\exp\left(-\delta_f(\mathbf{x},\eta)\sqrt{1+\left\|\nabla f(\mu)\right\|^2}\right) \quad (26)$$

since $\delta_f(x,y)=d_f(x,y)/\sqrt{1+\left\|\nabla f(y)\right\|^2}$.

In order to find the distribution induced by tBD $\delta_f(x, y)$, we consider a probability distribution of the form

$$p_\delta(\mathbf{x};\theta)=\tilde{h}(\mathbf{x})\exp\left(-\delta_f(\theta,\mathbf{x})-\tilde{g}(\theta)\right) \quad (27)$$

where $\tilde{g}$ is a normalization term depending only on θ. We call this the tBD distribution, since $\mathbf{x}$ are distributed according to the tBD values. This is not an exponential family but is a lifted exponential family. Given an exponential family (24), we define its lift by a curved exponential family

$$\tilde{p}(\mathbf{x};\theta)=\tilde{h}(\mathbf{x})\exp\left(\langle\tilde{\theta},\tilde{\mathbf{x}}\rangle-\tilde{g}(\theta)\right), \quad (28)$$

where

$$\tilde{\theta}=\begin{pmatrix} f(\theta) \\ \theta \end{pmatrix}, \quad \tilde{\mathbf{x}}=\frac{1}{w(\mathbf{x})}\begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \quad (29)$$

with $w(\mathbf{x})=\sqrt{1+\left\|\nabla f(\mathbf{x})\right\|^2}$. Here, $\tilde{h}(\mathbf{x})$ is adequately defined such that the integration of (28) converges and $\tilde{g}(\theta)$ corresponds to the normalization factor. We lifted $p(\mathbf{x}; \theta)$ to the space having one extra dimension and embed it (29) as a hyper-surface.

Theorem 4.1—Any tBD distribution corresponds to a lifted exponential family distribution.

It is easy to prove the theorem since $\delta_f(\theta, \mathbf{x})$ is written as

$$\delta_f(\theta, \mathbf{x}) = -\langle \tilde{\theta}, \tilde{\mathbf{x}} \rangle - \frac{f(\mathbf{x})}{w(\mathbf{x})}. \quad (30)$$

However, we may choose $\tilde{h}(\mathbf{x})$ adequately such that

$$\int \tilde{h}(\mathbf{x}) \exp\left(-\delta_f(\theta, \mathbf{x})\right) d\mathbf{x} \quad (31)$$

converges. A typical choice is Gaussian

$$\tilde{h}(\mathbf{x}) = c \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right). \quad (32)$$

We show that the arbitrariness of $\tilde{h}$ does not affect the stochastic inference given a data set $E = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$.

Theorem 4.2—The t-center of E is the Bayesian MAP estimator in the lifted exponential family (28) with a prior distribution

$$\pi(\theta) = \exp\left(-n\tilde{g}(\theta)\right). \quad (33)$$

Proof: The Bayesian MAP is the maximizer of

$$\log\left(\prod_i \pi(\theta) p_\delta(\mathbf{x}_i, \theta)\right) = \sum_i \log \tilde{h}(\mathbf{x}_i) - \sum_i \delta_f(\theta, \mathbf{x}_i), \quad (34)$$

which is the minimizer of $\Sigma \delta_f(\theta, \mathbf{x}_i)$.

When $f(\mathbf{x}) = \frac{\|\mathbf{x}\|^2}{2}$, (2) becomes a special radial basis function, whose general form is

$$\delta_f(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{2\sqrt{1 + \|\mathbf{x}_j\|^2}}\right). \quad (35)$$

Therefore, the tBD distribution is

$$p_\delta(\mathbf{x}; \theta) = \tilde{h}(\mathbf{x}) \exp\left(-\delta_f(\theta, \mathbf{x}) - \tilde{g}(\theta)\right), \quad (36)$$

where $\tilde{g}(\theta)$ is given from

$$\int p_\delta(\mathbf{x}; \theta) d\mathbf{x} = 1. \quad (37)$$

When we use the Bayesian prior $\pi(\theta) = \exp(g(\theta))$, the MAP for E is the maximizer of $\Pi_i \pi(\theta)p_\delta(\mathbf{x}_i; \theta)$, which is the minimizer of $\Sigma_i \delta_f(\theta, \mathbf{x}_i)$. In the next section, we use the t–center which is the result of the aforementioned minimization to develop a soft clustering algorithm.

5 tBD hard and soft clustering

When performing retrieval from a small database, it is possible to apply the brute-force search method by comparing the query shape with each shape in the database one by one. However, in the case of retrieval from a large database, it becomes impractical to use this brute-force search method because of the extremely high computational cost. To achieve real-time retrieval in a large database, we turn to a far more computationally efficient strategy, namely a *divide-and-conquer strategy*. First, utilizing the top down approach, we split the whole database into sub-clusters, and repeat recursively the same approach on the sub-clusters. Ideally, the divergence between shapes from the same cluster should be less than the divergences between images from different clusters, so that we choose a representative for each cluster, and assign each shape to the nearest cluster. There are two ways of assigning a shape to a cluster, assign it to a cluster deterministically, or assign the shape to a cluster according to some probability. The former corresponds to *hard clustering*, while the later case is *soft clustering*. These clustering algorithms were described in Banerjee et al. [45] for Bregman divergences (BDs), where the hard clustering chooses the centers of mass, and the soft clustering is shown to be equivalent to the celebrated Expectation-Maximization algorithm (EM) for learning mixtures of dual exponential families (EFs). In this paper, we consider tBDs instead of BDs and the tBD clustering algorithms are adapted accordingly. For hard clustering, the cluster centers are no longer centers of mass but barycenters with weights inversely depending on the norm of the gradient of the tBD generator. Similarly, the tBD soft clustering can be interpreted as the expectation-maximization algorithm for learning mixtures of **lifted** exponential families (lEFs) based on the bijection described in Section 4. We now summarize the soft clustering algorithm in the algorithm block below.

Algorithm 1

Total Bregman Soft Clustering Algorithm

Input: $X=\{x_i\}_{i=1}^N$, number of clusters c.

Output: $M = \{m_j\}_{j=1}$ and $Q = \{q_j\}_{j=1}$, m_j is the cluster center for the j^{th} cluster with probability q_j.

Initialization: Randomly choose c elements from X as M and set Q to the uniform probability.

repeat

{assign x_i to clusters}

for $i = 1$ to N **do**

for $j = 1$ to c **do**

$$q(j|x_i) \leftarrow \frac{q_j \exp(-\delta_f(m_j, x_i))}{\sum_{j'=1}^{c} q_{j'} \exp(-\delta_f(m_{j'}, x_i))}$$

end for

end for

{update cluster centers}

for $j = 1$ to c **do**

$q_j \leftarrow \frac{1}{N}\sum_{i=1}^{N} q(j|x_i)$

$m_j \leftarrow$ t-center for cluster j

end for

until The change of the results between two consecutive iterations is below some sensitivity threshold.

For clustering data sets, one has to first choose an appropriate divergence. Banerjee et al. [45] showed experimentally that the clustering result is best when the divergence is chosen according to the underlying generative process of data. That is, for a data set drawn from a mixture of exponential families, the hard clustering algorithm performs *experimentally* best if we choose the corresponding dual Bregman divergence [45] (p. 1737). The same experimental phenomenon holds for tBD clustering as described below. Further, we present comparative results for clustering using Bregman and total Bregman divergences on a synthetic data set. We see that the total Bregman divergences outperform the usual Bregman divergences in all experimental scenarios because of its inherent statistical robustness. In section 6 we further demonstrate that similar experimental results are obtained for shape retrieval applications on real-world data sets. We did four experiments using the same data sets as Banerjee et al. [45]. The first one is based on several 1D data sets of 300 samples each, generated from mixtures of Gaussian and Binomial models respectively. Both mixture models had three components with equal priors centered at 10, 20 and 40. The standard deviation of the Gaussian distribution was set to 5 and the number of trials of the Binomial distribution was set to 300 so as to make the two models somewhat similar to each other, in terms of the variance which is almost the same for all the models. We also use the same method to generate 2D and 5D data sets and compare the algorithms on them.

The accuracy of clustering was measured by using the normalized mutual information (NMI) [58], [59] between the predicted clusters and the original clusters that generated the samples, and the results were averaged over 30 trials.

Table 2 lists the NMI resulted from soft clustering using BD[1] and tBD. Gaussian mixture and Binomial mixture represent the models that generated the data sets. $d_{Gaussian}$ and $d_{Binomial}$ represent the Bregman divergences for Gaussian and Binomial distributions while $\delta_{Gaussian}$ and $\delta_{Binomial}$ represent the tBD for Gaussian and Binomial distributions. More specifically,

$$\delta_{Gaussian}(x_1, x_2) = \frac{f(x_1) - f(x_2) - \langle x_1 - x_2, \nabla f(x_2)\rangle}{\sqrt{1 + \left\|\nabla f(x_2)\right\|^2}} \quad (38)$$

$f(x) = \frac{1}{2}x^2$ and $q(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$. $\delta_{Binomial}$ is in the same format as (38) but $q(x) = \binom{n}{x} p^x (1-p)^{n-x}$, and p is the probability for a single success.

For Table 2, in ((a), (b), and (c)), rows 1 and 2 correspond to the NMI between the original and the predicted clusters obtained by applying the Bregman clustering algorithm using the Bregman divergences $d_{Gaussian}$ and $d_{Binomial}$ [45] respectively. Rows 3 and 4 correspond to

[1]For the implementation of BD soft clustering, we used the public domain jMEF library [55] from http://www.lix.polytechnique.fr/~nielsen/MEF/

the NMI yielded by the tBD clustering algorithm using $\delta_{Gaussian}$ and $\delta_{Binomial}$ respectively. The numbers in Table 2 demonstrate that using $\delta_{Gaussian}$ to cluster data sets generated by Gaussian mixture and using $\delta_{Binomial}$ to cluster data sets generated by Binomial mixture gives better NMI than using $d_{Gaussian}$ and $d_{Binomial}$. More importantly, using $\delta_{Gaussian}$ to measure the data sets generated by Binomial mixture gives much better NMI than using $d_{Gaussian}$ to perform the same task. This is also true with $\delta_{Binomial}$ and $d_{Binomial}$. This is very useful in the real applications, because often, one does not know the model that generates the data, and instead we have to blindly choose some divergence measure. But what we are sure now is that tBD is always better than Bregman divergence when using the same generating functions.

Remarks

From Table 2, we can see that with the increasing dimension, tBD soft clustering becomes more accurate than BD clustering, and the performance difference between tBD clustering and BD clustering becomes larger.

Also, if we fix the dimension of data and the original number of clusters c, and let the predicted cluster number $\tilde{c}$ approach to c, the NMI of tBD clustering increases faster than that of BD clustering. This is illustrated in Table 3. The data set for Table 3 is generated using the Gaussian generative model and the original number of clusters is $c = 3$. From Table 3, we can see tBD soft clustering behaves consistently better than BD soft clustering for which even the predicted number of clusters is incorrect. This is fundamental in image segmentation applications where it is necessary to partition an image into multiple regions or sets, and also can typically be used to locate objects and boundaries.

6 Shape retrieval using *t*-centers

The task of shape retrieval is to find the best match from a database of shapes given a query shape. In this section, we propose an efficient and accurate method for shape retrieval that includes an easy to use shape representation, and an analytical shape dissimilarity divergence measure. Also, we present an efficient scheme to solve the computationally expensive problem encountered when retrieving from a large database. The scheme is composed of clustering and efficient pruning, which will be elaborated on in Section 6.3.

6.1 Shape representation

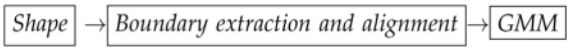
A time and space efficient shape representation is fundamental to shape retrieval. Given a segmented shape (or a binary image), we use a Gaussian Mixture Model (GMM) [2], [3], [4] to represent it. The procedure for obtaining the GMM from a shape is composed of three steps. First, we extract the points on the shape boundary or surface (to make it robust, for each point on the boundary, we also picked its closest 2 neighbors off the boundary), since MPEG-7 shapes are binary, the points that have nonzero gradient lie on the boundary (this step uses one line of MATLAB code). After getting the boundary points for every shape, we use the affine alignment proposed by Ho et al. [60] to align these points to remove the effect of rigid transformations (note that other alignment methods [22] will also be good). Given two sets of points $\{x_i\}_{i=1}^m$ and $\{y_j\}_{j=1}^n$ where $\{x_i\}$ are from a query and $\{y_j\}$ from the database, we can find affine alignment (A, b), $A \in \mathbf{GL}(2)$[2], $b \in \mathbb{R}^2$, such that $g(A, b) = \Sigma_i \min_j \{(Ax_i + b - y_j)^2\}$ achieves minimum, and then we use the aligned $\{\overline{x}_i | \overline{x}_i = Ax_i + b\}_{i=1}^m$ to represent the original point set $\{x_i\}_{i=1}^m$. This step is also very simple due to the explicit solution of (A, b), and it only takes several lines of MATLAB code to implement. Finally, we compute the

[2]**GL**(**2**): The set of 2 × 2 invertible matrices.

GMM from the aligned boundary points. A parametric GMM is a weighted combination of Gaussian kernels, which can be written as

$$p(x)=\sum_{i=1}^{m} a_i \mathcal{N}(x;\mu_i, \sum_i), 0 \le a_i \le 1, \sum_{i}^{n} a_i=1, \quad (39)$$

where m is the number of components; $\mathcal{N}(x; \mu_i, \Sigma_i)$ is the Gaussian density function with mean μ_i, variance Σ_i, and weight a_i in the mixture model. The mixture model is obtained by applying the EM algorithm and iteratively optimizing the centers and widths of the Gaussian kernels. The number m of components in the GMM model should be as small as possible, but makes the determinant of the covariance for each component not large (we found that m = 10 is a good compromise for MPEG-7 database). The above process is portrayed using the flow chart shown below.



Some concrete examples of the application of the flow chart are shown in Fig. 3.

6.2 Shape dissimilarity comparison using total square loss (tSL)

Total square loss (tSL) is a special class of tBD. As shown in Table 1, the generator function f on the scalar space for tSL is $f(x) = x^2$, and the tSL between two elements x and y is $tSL(x, y)=\frac{(x-y)^2}{\sqrt{1+4y^2}}$. Furthermore, tSL can be generalized to the space of vectors as well as functions. In our experiment, we use tSL to measure the difference between GMMs.

After getting the GMM representation of each shape, we use tSL to compare two GMMs, and take the difference as the dissimilarity between the corresponding shapes. Note that one could also use tKL to compare the difference between distributions. However, to compare the difference between GMMs, tSL gives closed form solution, but tKL does not and this motivates us to use tSL. Suppose two shapes have the following GMMs p_1 and p_2,

$$p_1(x)=\sum_{i=1}^{m} a_i^{(1)} \mathcal{N}(x;\mu_i^{(1)}, \sum_i^{(1)}), \quad (40)$$

$$p_2(x)=\sum_{i=1}^{m} a_i^{(2)} \mathcal{N}(x;\mu_i^{(2)}, \sum_i^{(2)}). \quad (41)$$

Since

$$\int \mathcal{N}(x;\mu_1, \sum_1)\mathcal{N}(x;\mu_2, \sum_2)dx=\mathcal{N}(0;\mu_1-\mu_2, \sum_1+\sum_2),$$

we can arrive at

$$tSL(p_1, p_2)=\frac{\int (p_1-p_2)^2 dx}{\sqrt{1+\int 4p_2^2 dx}}=\frac{d_1+d_2-d_{1,2}}{\sqrt{1+4d_2}}, \quad (42)$$

where

$$d_1=\sum_{i,j=1}^{m} a_i^{(1)} a_j^{(1)} \mathcal{N}(0;\mu_i^{(1)}-\mu_j^{(1)}, \sum_j^{(1)}+\sum_j^{(1)}), \quad (43)$$

$$d_2=\sum_{i,j=1}^{m} a_i^{(2)} a_j^{(2)} \mathcal{N}(0;\mu_i^{(2)}-\mu_j^{(2)}, \sum_j^{(2)}+\sum_j^{(2)}), \quad (44)$$

$$d_{1,2}=2\sum_{i,j=1}^{m} a_i^{(1)} a_j^{(2)} \mathcal{N}(0;\mu_i^{(1)}-\mu_j^{(2)}, \sum_i^{(1)}+\sum_j^{(2)}), \quad (45)$$

$$\mathcal{N}(0;\mu, \sum)=\frac{1}{(\sqrt{2\pi})^e \sqrt{\det(\sum)}}\exp\left(-\frac{1}{2}\mu' \sum^{-1} \mu\right), \quad (46)$$

and e is the dimension of μ. Given a set of GMMs $\{p_l\}_{l=1}^n$, $p_l=\sum_{i=1}^m a_i^{(l)} \mathcal{N}(x;\mu_i^{(l)}, \sum_i^{(l)})$ its t-center can be obtained from equation (14), which is

$$\begin{aligned} \overline{p}&=\frac{\sum_{l=1}^n w_l p_l}{\sum_{l=1}^n w_l}, \\ w_l&=(1+4d_l)^{-1/2}, \\ d_l&=\sum_{i,j=1}^{m} a_i^{(l)} a_j^{(l)} \mathcal{N}(0;\mu_i^{(l)}-\mu_j^{(l)}, \sum_i^{(l)}+\sum_j^{(l)}). \end{aligned} \quad (47)$$

We evaluate the dissimilarity between the GMM of the query shape and the GMMs of the shapes in the database using *tSL*, and the smallest dissimilarities correspond to the best matches.

6.3 Shape retrieval in MPEG-7 database

The proposed divergence is evaluated on the shape retrieval problem using the MPEG-7 database [42], which consists of 70 different objects with 20 shapes per object, for a total of 1400 shapes. This is a fairly difficult database to perform shape retrieval because of its large intraclass variability, and, for many classes, small interclass dissimilarity, and furthermore, there are missing parts and occlusions in many shapes.

We cluster the database into hierarchical clusters, calculate their t-centers and compare the query shape with the t-centers hierarchically. For the clustering part, we applied both hard clustering and soft clustering.

For hard clustering, we apply a variation of k-tree method by setting $k = 10$ at the first level of clustering, 7 at the second level, 5 at the third level and 2 at all following levels, so the average number of shapes in each cluster is 140, 20, 4, 2, and 1.

In the k-tree, every key is represented by a mixture of Gaussians, every inner node (including the root) has 1 to k keys, each of which is the t-center of all keys in its child nodes, and the key for a leaf node is a mixture of Gaussians for an individual shape. The k-tree illustration is shown in Fig. 4.

During retrieval, we only need to compare the query with the representatives, and once the best match representative is obtained, we compare the query with the t-centers of the relative sub-clusters, recursively doing so, until the required number of best matches are found.

For soft clustering, we append a semi-hard assignment to the soft clustering algorithm, i.e., after the soft clustering converges, we will assign the shape x_i to cluster C_j, if $p(C_j|x_i) \geq a$. We use $a = 1/2$, so that one shape can be assigned to at most 2 clusters at the tree leaf level, but a cluster center may be dependent on x_i even though x_i is not assigned to this cluster finally.

The clustering process is a coarse to fine procedure, which greatly enhances efficiency while guaranteeing accuracy. Also, we compare the clustering accuracy of *tSL*, $\chi 2$ [3] and *SL* soft and hard clustering by a reasonable measure, which is the optimal number of categories per cluster (denoted by $|C|^*$, $|C|$ represents the cardinality of C, i.e., the number of categories in C) divided by the average number of categories in each cluster (denoted by $Avg(|C|)$). For example, at the first level clustering, there are 10 clusters $\{C_i\}_{i=1}^{10}$, with an average of 140 shapes per cluster, and thus, the optimal number of categories per cluster $|C|^* = 140/20 = 7$; $Avg(|C|)=\frac{\sum_{i=1}^{10}|C_i|}{10}$. The smaller the number of categories per cluster, the higher is the clustering accuracy, and the more accurate will be the categories separation. Note the optimal clustering accuracy is 1. Fig. 5 compares the clustering accuracy of *tSL*, χ^2 and *SL* soft and hard clustering, which shows that *tSL* soft clustering has a striking clustering accuracy, implying substantial capability to detect outliers, occlusion, missing parts, and strong ability to distinguish shapes from different categories.

We include here several groups of retrieval results in Fig. 6, which show that our method can deal very well with scale, rotation, pose, occlusion, missing parts, great intraclass dissimilarity and large interclass similarity.

The evaluation of accuracy for retrieval in the whole MPEG-7 database is based on the well recognized criterion, recognition rate [5], [9], [61], [42]. Each shape is used as a query and the top 40 matches are retrieved from all 1400 shapes. The maximum possible number of correct retrievals for each query is 20, and hence there are a total of 28,000 possible matches with the recognition rate reflecting the number of correct matches divided by this total.

Table 4 lists the recognition rate we obtained and comparisons with some of the existing techniques. Note that our method gives high recognition rate, even though it is not as good as [15], [62], however, our method does not need any preprocessing of the shapes or any post-processing of the similarities. Simplicity and efficiency of our method are the key salient features distinguishing it from other methods.

6.4 Brown database

Additionally, we apply our proposed method to the Brown database [14], which contains 9 shape categories, where each shape category has 11 different segmented binary shapes and 99 shapes in total. We use GMM to represent each shape, the number of components for each GMM is decided using the same way as in the MPEG-7 experiment, and compare the difference of shapes using the tSL between their corresponding GMMs. We tested our method using the criteria as in [8], [61], [14], [15], [16]: every shape is taken as the query, and compared with all the shapes in the database. We then find the best 10 matches, and check the number of correct matches, i.e., the number of shapes which belong to the same

[3] $\chi^2: d(p,q)=\int_x \frac{(p(x)-q(x))^2}{(p(x)+q(1))^2}dx$

category as the query shape. This process is repeated by taking each one of the 99 shapes in the whole data set as the query shape. Then we check the total correct matches for the i^{th} found shapes, $i = 1, 2, \cdots, 10$ (the maximum number of correct matches is 99), which are shown in Table 5. We can see that our method performs quite well.

6.5 Swedish Leaf Data Set

The Swedish leaf data set [43] contains isolated leaves from 15 different Swedish tree species, with 75 leaves per species, and 1125 shapes in total. We use the classification criteria as in [7], [15], [19], [61], which used the 1-nearest-neighbor approach to measure the classification performance. For each leaf species, 25 samples are selected as a template and the other 50 are selected as targets. We use GMM to represent each shape, and use tBD soft clustering algorithm to cluster the shapes into different clusters. Shape classification results on this data set are shown in Table 6, from which we can see that our accessible shape representation plus tBD soft clustering algorithm yields very good results.

6.6 3D Princeton Shape Benchmark

Our method performed very well in the domain of 2D shape retrieval and it can be extended very easily to higher dimensional space. Here, we evaluate our method on the Princeton Shape Benchmark (PSB) [44] containing 1814 3D models, which is divided into the training set (907 models in 90 classes) and the testing set (907 models in 92 classes). We evaluate our method on the testing set, and compare our results with others in three ways, Nearest Neighbor (NN), Discounted Cumulative Gain (DCG) and Normalized DCG (NDCG) using the software provided in PSB [44]. Our method outperforms the other methods when using NN criteria, and can find the first closest matches that belong to the query class more accurately.

7 Conclusions and future work

In this paper, we defined and investigated properties of total Bregman divergences (tBD). We introduced the tBD-based ℓ_p-norm centers, and report closed-form expressions for the ℓ_1-norm t-center that is provably robust to outliers. We extend the work of Banerjee et. al [45] by proposing both tBD hard and soft clustering algorithms that exhibit experimental robustness compared to conventional Bregman divergences. We also developed a simple and efficient shape retrieval approach, using shape alignment to remove rigid motion effects (translation, rotation, scale), then using a GMM to represent shape boundaries (contour or boundary surface) followed by an application of the tBD clustering algorithm to cluster the shapes into sub-clusters and store the shapes using a k-tree. The clustering has low computational cost, because the cluster center is in closed form and is only dependent on the GMM means and variances. The k-tree makes retrieval very efficient which takes logarithmic comparisons. Furthermore, each comparison is very fast because the tBD between two GMMs also has an explicit form. In summary, our method is efficient, invariant to rigid transformations, robust to outliers, and yields better or similar results compared with the state-of-the-art techniques.

For future work, we plan to apply our method to real-world image data set classification and video retrieval, with the existence of clutter and motion, which will be more challenging and interesting.

Acknowledgments

This research was in part supported by the NIH grant EB007082 to Vemuri and the University of Florida Fellowship to Liu.

References

1. Liu M, Vemuri BC, Amari SI, Nielsen F. Total Bregman divergence and its applications to shape retrieval. IEEE CVPR. 2010:3463–3468.
2. Chui H, Rangarajan A. A feature registration framework using mixture models. IEEE Workshop on Mathematical Methods in Biomedical Image Anal. 2000:190–197.
3. Jian B, Vemuri BC. A robust algorithm for point set registration using mixture of gaussians. IEEE ICCV. 2005; 2:1246–1251.
4. Jian B, Vemuri BC. Robust point set registration using gaussian mixture models. IEEE Trans Pattern Anal Mach Intell. 2011; 33(8):1633–1645.
5. Peter A, Rangarajan A, Ho J. Shape L'Ane rouge: Sliding wavelets for indexing and retrieval. IEEE CVPR. 2008:1–8.
6. Ling H, Jacobs D. Using the inner-distance for classification of articulated shapes. IEEE CVPR. 2005; 2:719–726.
7. Ling H, Okada K. An efficient earth movers distance algorithm for robust histogram comparison. IEEE TPAMI. 2007; 29:840–853.
8. Bai X, Yang X, Latecki LJ, Liu W, Tu Z. Learning context sensitive shape similarity by graph transduction. IEEE TPAMI. 2010; 32(5):861–874.
9. McNeill G, Vijayakumar S. Hierarchical procrustes matching for shape retrieval. IEEE CVPR. 2006; 1:885–894.
10. Biederman I, Ju G. Surface vs. edge-based determinants of visual recognition. Cognitive Psychology. 1988; 20:38–64. [PubMed: 3338267]
11. Gavrila DM. A Bayesian, exemplar-based approach to hierarchical shape matching. IEEE TPAMI. 2007:1408–1421.
12. Macrini D, Siddiqi K, Dickinson S. From skeletons to bone graphs: Medial abstraction for object recognition. IEEE CVPR. 2008:1–8.
13. Weinland D, Boyer E. Action recognition using exemplar-based embedding. IEEE CVPR. 2008:1–7.
14. Sebastian TB, Klein PN, Kimia BB. Recognition of shapes by editing their shock graphs. IEEE TPAMI. 2004; 26(5):550–571.
15. Temlyakov A, Munsell BC, Waggoner JW, Wang S. Two perceptually motivated strategies for shape classification. IEEE CVPR. 2010:2289–2296.
16. Tu Z, Yuille A. Shape matching and recognition: Using generative models and informative features. ECCV. 2004; 3:195–209.
17. Xu C, Liu J, Tang X. 2D shape matching by contour flexibility. IEEE TPAMI. 2009; 31(1):180–186.
18. Gopalan R, Turaga P, Chellappa R. Articulation-invariant representation of non-planar shapes. ECCV. 2010; 6313:286–299.
19. Yang X, Tezel SK, Latecki L. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. IEEE CVPR. 2009:357–364.
20. Ling H, Yang X, Latecki LJ. Balancing deformability and discriminability for shape matching. ECCV. 2010:411–424.
21. Yang X, Latecki LJ. Affinity learning on a tensor product graph with applications to shape and image retrieval. IEEE CVPR. 2011:2873–2880.
22. Munsell BC, Temlyakov A, Styner M, Wang S. Pre-organizinig shape instances for landmark-based shape correspondence. IJCV. 2011
23. Siddiqi, K.; Pizer, SM. Medial representations: Mathematics, Algorithms and Applications. Springer; 2008.
24. Leyton M. A process-grammar for shape. Artificial Intelligence. 1988; 34:213–247.
25. Flynn PJ, Jain A. CAD-based computer vision: From CAD models to relational graphs. IEEE TPAMI. 1991; 13:114–132.
26. Ferrucci V, Paoluzzi A. Extrusion and boundary evaluation for multidimensional polyhedra. Computer Aided Design. 1991; 1:40–50.

27. Damski JC, Gero JS. A logic-based framework for shape representation. Computer-Aided Design. 1996; 28(3):169–181.
28. Pobil APD, Serna MA. Spatial representation and motion planning. Lecture Notes in Computer Science, Springer-Verlag. 1995; (1014):169–181.
29. Maji S, Berg AC, Malik J. Classification using intersection kernel support vector machines is efficient. IEEE CVPR. 2008:1–8.
30. Joshi SH, Klassen E, Srivastava A, Jermyn I. A novel representation for riemannian analysis of elastic curves in $\mathbb{R}^n$. IEEE CVPR. 2007:1–7.
31. Younes L. Computable elastic distance between shapes. SIAM Journal of Applied Mathematics. 1998; 58(2):565–586.
32. Chen L, Feris R, Turk M. Efficient partial shape matching using smith-waterman algorithm. IEEE CVPRW. 2008:1–6.
33. Shotton J, Blake A, Cipolla R. Multiscale categorical object recognition using contour fragments. IEEE TPAMI. 2008; 30:1270–1281.
34. Opelt A, Pinz A, Zisserman A. A boundary-fragment-model for object detection. ECCV. 2006; 2:575–588.
35. Ferrari V, Tuytelaars T, Gool LV. Object detection by contour segment networks. ECCV. 2006; 3:14–28.
36. Mori G, Belongie S, Malik J. Efficient shape matching using shape contexts. IEEE TPAMI. 2005; 27:1832–1837.
37. Ding L, Belkin M. Component based shape retrieval using differential profiles. ACM Int Conf Multimedia Infor Retrieval. 2008:216–222.
38. Vemuri BC, Liu M, Amari S-I, Nielsen F. Total Bregman divergence and its applications to DTI analysis. IEEE Transactions on Medical Imaging (TMI). 2011; 30(2):475–483.
39. Groen PD. An introduction to total least squares. 1996; 14:237–253.
40. Lancaster J, et al. k-Tree method for high-speed spatial normalization. Human Brain Mapping. 1998:358–363. [PubMed: 9788072]
41. Nielsen F, Piro P, Barlaud M. Bregman vantage point trees for efficient nearest neighbor queries. IEEE Int Conf Multimedia & Expo. 2009:878–881.
42. Latecki LJ, Lakamper R, Eckhardt U. Shape descriptors for non-rigid shapes with a single closed contour. IEEE CVPR. 2000; 1:424–429.
43. Soderkvist, OJO. Computer vision classification of leaves from swedish trees. Linkoping University; Sweden: 2001.
44. Shilane P, Min P, Kazhdan M, Funkhouser T. The Princeton shape benchmark. Proc SMI. 2004:167–178.
45. Banerjee A, Merugu S, Dhillon IS, Ghosh J. Clustering with Bregman divergences. J Mach Learn Res. 2005; 6:1705–1749.
46. Zhang J. Divergence function, duality, and convex analysis. Neural Computation. 2004; 16:159–195. [PubMed: 15006028]
47. Frigyik BA, Srivastava S, Gupta MR. Functional Bregman divergence. Int Symp Inf Theory. 2008; 54:5130–5139.
48. Nielsen F, Boltz S. The Burbea-Rao and Bhattacharyya centroids. IEEE Trans Inf Theory. 2011
49. Amari S. Integration of Stochastic Models by Minimizing α-Divergence. Neural Computation. 2007; 19(10):2780–2796. [PubMed: 17716012]
50. Nielsen F, Nock R. On the smallest enclosing information disk. Inf Process Lett. 2008; 105:93–97.
51. Wainwright MJ, Jordan MI. Graphical models, exponential families and variational inference. Foundations and Trends in Machine Learning. 2008
52. Collins M, Schapire R, Singer Y. Logistic regression, adaboost, and bregman distances. Machine Learning. 2001; 48(1–3):253–285.
53. Pietra SD, Pietra VD, Lafferty J. Duality and auxiliary functions for bregman distances. Technical Report CMU-CS-01-109, School of Computer Science, CMU. 2001
54. Cover, TM.; Thomas, J. Elements of Information Theory. Wiley Interscience; 1991.

55. Nielsen, F.; Garcia, V. Statistical exponential families: A digest with flash cards. 2009. arXiv.org: 0911.4863
56. Amari, S.; Nagaoka, H. Methods of Information Geometry. American Mathematical Society; 2001.
57. Barndorff-Nielsen, O. Information and exponential families in statistical theory. Wiley; New York: 1978.
58. Strehl A, Ghosh J. Cluster ensembles - a knowledge reuse framework for combining partitionings. Journal of Machine Learning Research. 2002; 3(3):583–617.
59. Manning, CD.; Raghavan, P.; Schütze, H. Introduction to information retrieval. Cambridge University Press; 2008.
60. Ho J, Yang M, Rangarajan A, Vemuri B. A new affine registration algorithm for matching 2D point sets. Proceedings of the Eighth IEEE Workshop on Appl of Comput Vis. 2007; 89:25–30.
61. Felzenszwalb P, Schwartz J. Hierarchical matching of deformable shapes. IEEE CVPR. 2007:1–8.
62. Bai X, Wang B, Wang X, Liu W, Tu Z. Co-transduction for shape retrieval. ECCV. 2010:861–874.
63. Sebastian TB, Klein PN, Kimia BB. On aligning curves. IEEE TPAMI. 2003; 25(1):116–125.
64. Belongie S, Malik J, Puzicha J. Shape matching and object recognition using shape contexts. IEEE TPAMI. 2002; 24:509–522.
65. Mokhtarian F, Abbasi S, Kittler J. Efficient and robust retrieval by shape content through curvature scale space. Int W Image Databases and Multi-Media Search. 1997:51–58.
66. Kontschieder P, Donoser M, Bischof H. Beyond pairwise shape similarity analysis. ACCV. 2009:655–666.
67. Papadakis S, Pratikakis I, Theoharis T. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. Pattern Recognition. 2007; 40(9):2437–2452.
68. Akgul CB, Sankur B, Yemez Y, Schmitt F. 3D model retrieval using probability density-based shape descriptors. IEEE TPAMI. 2009; 31(6)
69. Vranic, D. Tools for 3D model retrieval. 2005. http://merkur01.inf.unikonstanz.de/3Dtools/
70. Osada R, Funkhouser T, Chazelle B, Dobkin D. Shape distributions. ACM Trans Graphics. 2002; 21(4):807–832.

Biographies

Meizhu Liu received her B.S. degree from the University of Science and Technology of China, Hefei, Anhui, China in July 2007. She received her Ph.D. degree from the University of Florida, Gainesville, Florida in December 2011. Her research interests include computer vision, machine learning, medical image processing and image analysis, information theory and computational geometry. She is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Society for Industrial and Applied Mathematics (SIAM).

Baba C. Vemuri received his Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin in 1987. He joined the Department of Computer and Information Sciences at the University of Florida, Gainesville, FL in 1987, and is currently a university research foundation professor of Computer and Information Sciences and Engineering. He was a program chair for the 11th IEEE International Conference on Computer Vision (ICCV 2007). He has been an area chair and a program committee member of several IEEE, MICCAI, and IPMI conferences. He served as an associate editor for several journals, including the IEEE TPAMI (1992–96), IEEE TMI (1997–2003) and CVIU (2000–11). He is currently an associate editor for the Journal of Medical Image Analysis. His research interests include medical image analysis, computational vision, information geometry, machine learning, and applied mathematics. For the last decade, his research work has primarily focused on information geometric methods applied to computer vision and medical image analysis. He has published more than 150 refereed articles in journals and conference proceedings and has received several best paper awards to date. He is a fellow of the ACM and the IEEE.

Shun-Ichi Amari received his Dr. Eng. degree from the University of Tokyo in 1963. He worked as an Associate Professor at Kyushu University and the University of Tokyo, and then a Full professor at the University of Tokyo, and is now Professor-Emeritus. He served as Director of RIKEN Brain Science Institute for five years, and is now its senior advisor. He has been engaged in research in wide areas of mathematical engineering, in particular, mathematical foundations of neural networks, including statistical neurodynamics, dynamical theory of neural fields, associative memory, self-organization, and general learning theory. Another main subject of his research is information geometry initiated, which provides a new powerful method to information sciences and neural networks. Dr. Amari served as President of Institute of Electronics, Information and Communication Engineers, Japan and President of International Neural Networks Society. He received Emanuel A. Piore Award and Neural Networks Pioneer Award from IEEE, the Japan Academy Award, Gabor Award from INNS, Caianiello Award, Bosom Friend Award from Chinese Neural Networks Council and C&C award, among many others.

Frank Nielsen is a researcher of the Fundamental Research Laboratory of Sony Computer Science Laboratories (Sony CSL FRL for short). He received his Bachelor and Master degree in computer science from École Normale Superieure (ENS) of Lyon, France in 1992 and 1994 respectively. He got his PhD from the University of Nice (France) in 1996. Then he served army as a scientific member in the computer science laboratory of École Polytechnique (LIX). In 1998, he joined Sony Computer Science Laboratories, Tokyo (Japan). His current research interests include computational geometry, vision, graphics, optimization and learning. Last but not least, he is interested in technology transfers that bring a complementary view of solving requirements in an industrial setting. He is an ACM member and a senior IEEE member.

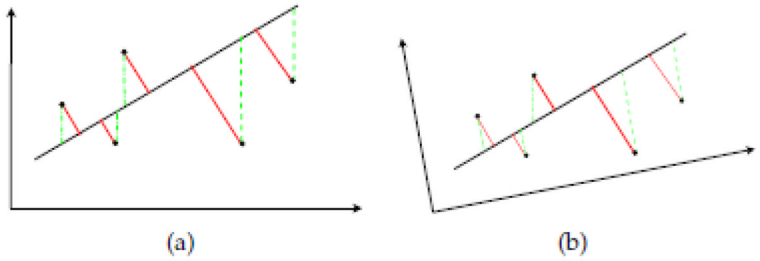


Fig. 1.

In each figure, the dotted green line is least squares, and the bold red line is total least squares, and the two black orthogonal lines indicate the coordinate system. (a) shows least squares and total least squares before rotating the coordinate system. (b) shows least squares and total least squares after rotating the coordinate system.

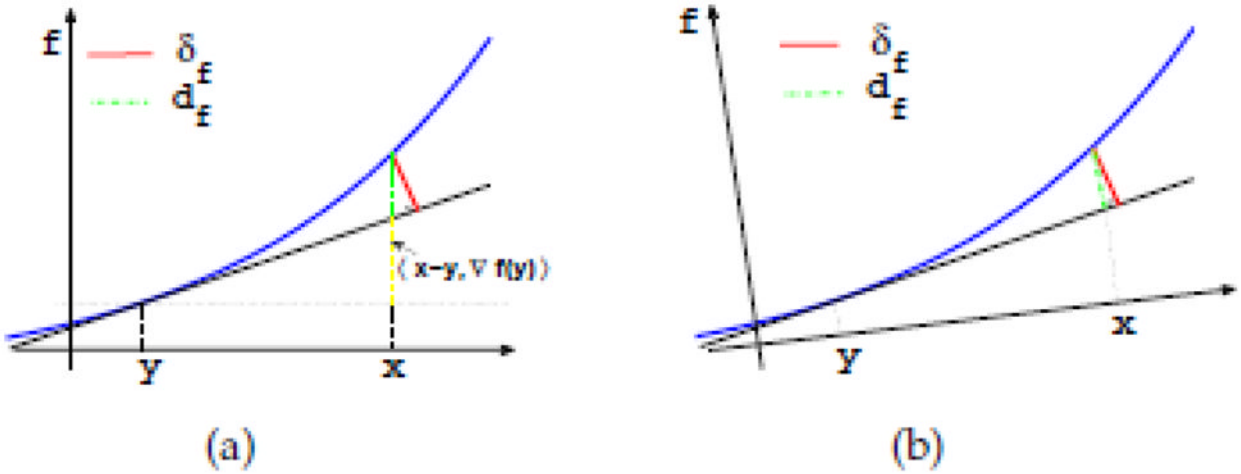


Fig. 2.

In each figure, $d_f(x, y)$ (dash dotted green line) is BD, $\delta_f(x, y)$ (bold red line) is tBD, and the two orthogonal arrows indicate the coordinate system. The length of the yellow dash line is $\langle x - y, \nabla f(y) \rangle$. (a) shows $d_f(x, y)$ and $\delta_f(x, y)$ before rotating the coordinate system. (b) shows $d_f(x, y)$ and $\delta_f(x, y)$ after rotating the coordinate system. Note that $d_f(x, y)$ changes with rotation unlike $\delta_f(x, y)$ which is invariant to rotation.

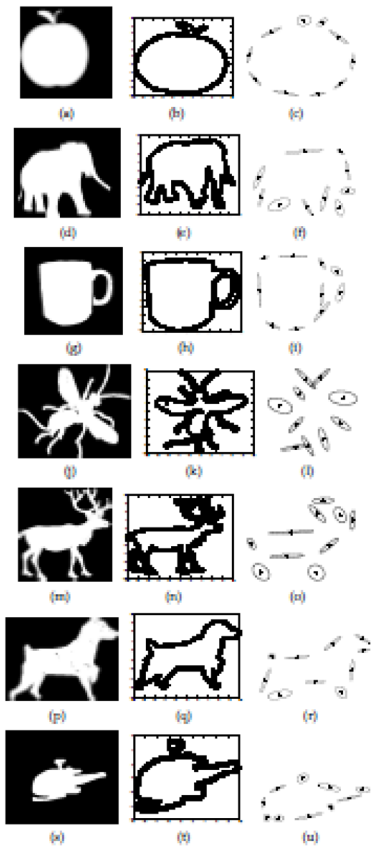


Fig. 3.

Left to right: original shapes; aligned boundaries; GMM with 10 components, the dot inside each ellipse is the mean of the corresponding Gaussian density function, and the transverse as well as the conjugate diameter for each ellipse correspond to the eigen values of the covariance matrix.

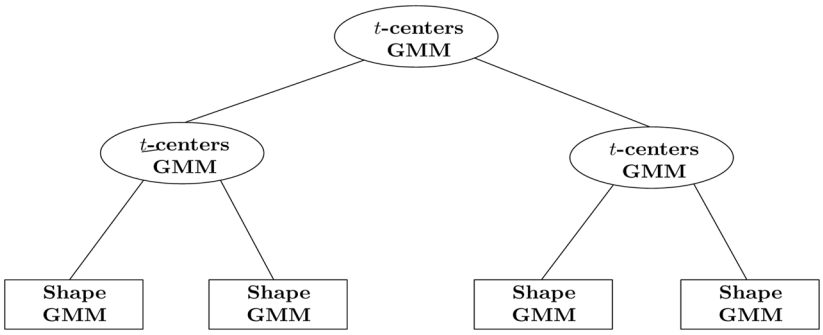


Fig. 4.

k-tree diagram. GMM: Gaussian mixture model. Each key is a mixture of Gaussians as explained in the experimental part. Each key in the inner nodes is the *t*-center of all keys in its child nodes. The key of a leaf is a mixture of Gaussians corresponding to an individual shape.

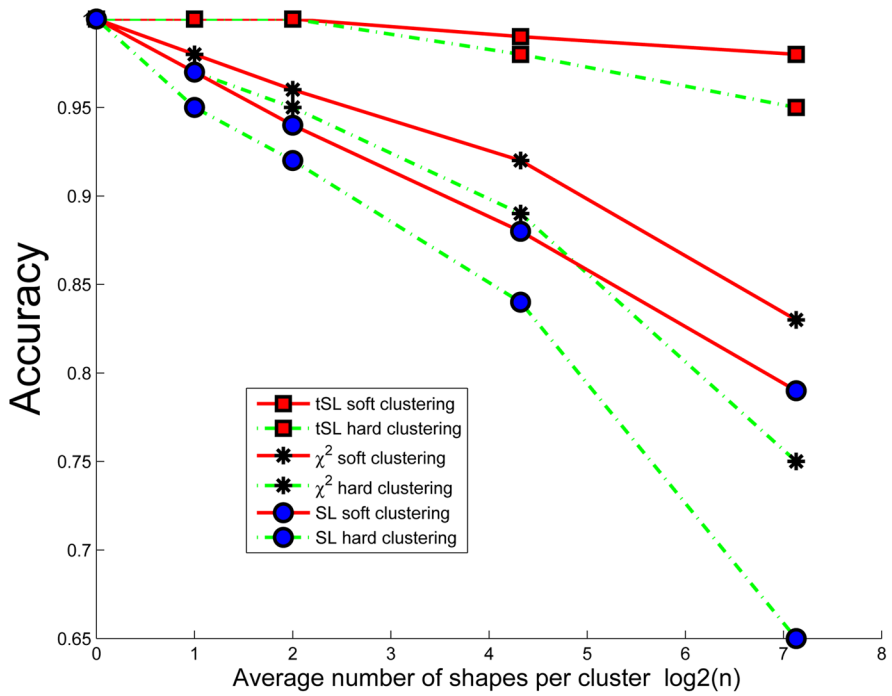


Fig. 5.

Comparison of clustering accuracy of *tSL*, χ^2 and *SL*, versus average number of shapes per cluster.

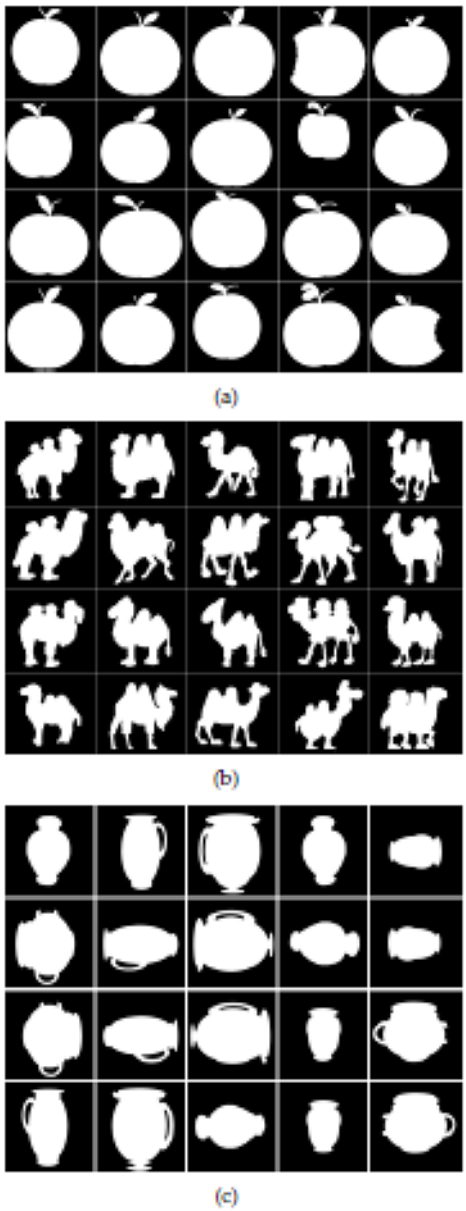


Fig. 6.
Retrieval results using our proposed method: the first shape in each figure is the query, and the other shapes are shown from left to right, up to down according to the ascending order of the divergence to the query.

TABLE 1

TBD δ_f and the corresponding t-center. $\dot{x} = 1 - x$, $\dot{y} = 1 - y$. c is the normalization constant to make it a pdf, $w_i = \frac{1/\sqrt{1+\|\nabla f(x_i)\|^2}}{\sum_j 1/\sqrt{1+\|\nabla f(x_j)\|^2}}$.

X	$f(x)$	$\delta_f(x, y)$	t-center	ℓ_1-norm BD center	Remark
$\mathbb{R}$	x^2	$\frac{(x-y)^2}{\sqrt{1+4y^2}}$	$\Sigma_i w_i x_i$	$\Sigma_i x_i$	total square loss (tSL)
$\mathbb{R}-\mathbb{R}_-$	$x \log x$	$\frac{x\log\frac{x}{y}+\bar{x}\log\frac{\bar{x}}{\bar{y}}}{\sqrt{1+y(1+\log y)^2+\bar{y}(1+\log\bar{y})^2}}$	$\Pi_i(x_i)^{w_i}$	$\Sigma_i x_i$	
$[0,1]$	$-\log x$	$\frac{\frac{x}{y}-\log\frac{x}{y}-1}{\sqrt{1+y^{-2}}}$	$\frac{\sum_i(x_i/(1-x_i))^{w_i}}{1+\sum_i(x_i/(1-x_i))^{w_i}}$	$\Sigma_i x_i$	total logistic loss
$\mathbb{R}_+$	$-\log x$	$\frac{\frac{x}{y}-\log\frac{x}{y}-1}{\sqrt{1+y^{-2}}}$	$\frac{1}{\sum_i w_i/x_i}$	$\Sigma_i x_i$	total Itakura-Saito distance
$\mathbb{R}$	e^x	$\frac{e^x-e^y-(x-y)e^y}{\sqrt{1+e^2y}}$	$\Sigma_i w_i x_i$	$\Sigma_i x_i$	
$\mathbb{R}^d$	$\|x\|^2$	$\frac{\|x-y\|^2}{\sqrt{1+4\|y\|^2}}$	$\Sigma_i w_i x_i$	$\Sigma_i x_i$	total squared Euclidean
$\mathbb{R}^d$	$x^t Ax$	$\frac{(x-y)^t A(x-y)}{\sqrt{1+4\|Ay\|^2}}$	$\Sigma_i w_i x_i$	$\Sigma_i x_i$	total Mahalanobis distance
Δ^d	$\sum_{j=1}^d x_j \log x_j$	$\frac{\sum_{j=1}^d x_j\log\frac{x_j}{y_j}}{\sqrt{1+\sum_{j=1}^d y_j(1+\log y_j)^2}}$	$c\Pi_i(x_i)^{w_i}$	$\Sigma_i x_i$	total KL divergence (tKL)

X	$f(x)$	$\delta_f(x, y)$	t-center	ℓ_q-norm BD center	Remark
$\mathbb{C}^{m\times n}$	$\|\|x\|\|_F^2$	$\frac{\|\|x-y\|\|_F^2}{\sqrt{1+4\|\|y\|\|_F^2}}$	$\sum_i w_i x_i$	$\sum_i x_i$	total squared Frobenius

TABLE 2

(a), (b) and (c) present the clustering results for the 1D, 2D and 5D data sets. Columns 2–3 correspond to the NMI between the original and predicted clusters (original number of clusters is 3, predicted number of clusters is 5) obtained by applying the Bregman soft clustering algorithm corresponding to the $d_{Gaussian}$ and $d_{Binomial}$ and tBD soft clustering algorithm corresponding to $\delta_{Gaussian}$ and $\delta_{Binomial}$ respectively.

(a)

Measure	Gaussian Mixture	Binomial Mixture
$d_{Gaussian}$	0.73660±0.00142	0.64998±0.00035
$d_{Binomial}$	0.64307±0.00066	0.72982±0.00379
$\delta_{Gaussian}$	**0.75076±0.00193**	**0.65089±0.00018**
$\delta_{Binomial}$	**0.67899±0.00773**	**0.74450±0.00218**

(b)

Measure	Gaussian Mixture	Binomial Mixture
$d_{Gaussian}$	0.43922±0.03577	0.38246±0.05020
$d_{Binomial}$	0.36805±0.05513	0.42987±0.04747
$\delta_{Gaussian}$	**0.59385±0.11910**	**0.50518±0.07280**
$\delta_{Binomial}$	**0.52136±0.09187**	**0.57063±0.05173**

(c)

Measure	Gaussian Mixture	Binomial Mixture
$d_{Gaussian}$	0.39869±0.00049	0.38246±0.05020
$d_{Binomial}$	0.19765±0.00025	0.14205±0.05619
$\delta_{Gaussian}$	**0.52360±0.00018**	**0.41516±0.05320**
$\delta_{Binomial}$	**0.36883±0.00035**	**0.52164±0.04173**

TABLE 3

NMI between the original clusters and the clusters obtained from tBD and BD soft clustering algorithms respectively. $\hat{c}$) is the predicted number of clusters. The original number of clusters is 3

$\hat{c}$	$d_{Gaussian}$	$\delta_{Gaussian}$
10	0.65627±0.00020	0.656364±0.00034
8	0.68417±0.00122	0.68554±0.00109
6	0.69958±0.00112	0.70202±0.00154
5	0.73660±0.00142	0.75076±0.00193
3	0.86115±0.00141	**0:98658 ± 0:00120**

TABLE 4

Recognition rates for shape retrieval in the MPEG-7 database. Soft clustering using total Bregman tSL divergence performs well, gaining 16% increase over ordinary Bregman SL divergence.

Technique	Recognition rate (%)
GMM+soft clustering+ tSL	**93.41**
GMM+soft clustering+SL	76.48
GMM+hard clustering+tSL [1]	89.1
GMM+hard clustering+SL [1]	65.92
Shape-tree [61]	87.7
IDSC + DP + EMD [7]	86.56
Hierarchical Procrustes [9]	86.35
IDSC + DP [6]	85.4
Shape L'Âne Rouge [5]	85.25
Generative Models [16]	80.03
Curve Edit [63]	78.14
SC + TPS [64]	76.51
Visual Parts [42]	76.45
CSS [65]	75.44
Perceptual Strategy + IDSC + LCDP [15]	95.60
IDSC + Mutual Graph [66]	93.4
IDSC + LCDP + unsupervised GP [19]	93.32
IDSC + LCDP [19]	92.36
IDSC + LP [8]	91.61
Contour Flexibility [17]	89.31
Perceptual Strategy + IDSC [15]	88.39
SC + IDSC + Co-Transduction [62]	97.72
AIR[18]	93.67
ASC + LCDP [20]	95.96
AIR + DN + TPG Diffusion [21]	99.99

TABLE 5

Recognition rates for shape retrieval from the Brown database. Using total Bregman (tSL) divergence, we obtain the best performance.

Technique	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
GMM+tSL	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**99**
GMM+SL	99	99	97	95	87	91	92	89	83	69
Perceptual Strategy + IDSC + LCDP [15]	99	99	99	99	99	99	99	99	99	99
IDSC + LP [8]	99	99	99	99	99	99	99	99	97	99
Shape-tree [61]	99	99	99	99	99	99	99	97	93	−86
IDSC [7]	99	99	99	98	98	97	97	98	94	79
Shock-Graph Edit [14]	99	99	99	98	98	97	96	95	93	82
Generative Models [16]	99	97	99	98	96	96	94	83	75	48

TABLE 6

Recognition rates for shape retrieval from the Swedish leaf database.

Technique	recognition rate(%)
GMM+soft clustering + tSL	**98.33**
GMM+hard clustering + tSL	97.92
GMM+soft clustering+SL	94.21
GMM+hard clustering+SL	90.89
Perceptual Strategy + IDSC + LCDP [15]	98.27
IDSC + LCDP [19]	98.20
Shape-tree [61]	96.28
IDSC [7]	94.13

TABLE 7

Retrieval comparison with other methods (CRSP [67], DSR [68], DBF [68], DBI [69], SIL [69], D2 [70]) on PSB. Observe that performance drops significantly if we consider ordinary Bregman divergences (BD) instead of total Bregman divergences (tBD).

	tBD	**BD**	**CRSP**	**DSR**	**DBF**	**DBI**	**SIL**	**D2**
NN	**72.3**	53.4	67.9	66.5	68.6	60.9	55.7	31.1
DCG	**66.7**	46.9	66.8	66.5	65.9	61.4	59.7	43.4
NDCG	**16.1**	4.5	16.4	15.9	14.9	7	4.1	−24.4