Maximum Normalized Spacing for Efficient Visual Clustering

Zhi-Gang Fan
Advanced R&D Center
SHARP Electronics
(Shanghai) Co. Ltd
1387 Zhangdong Road
Shanghai, China
zhigang.fan@cn.sharp-world.com

Yadong Wu
Advanced R&D Center
SHARP Electronics
(Shanghai) Co. Ltd
1387 Zhangdong Road
Shanghai, China

Bo Wu
Advanced R&D Center
SHARP Electronics
(Shanghai) Co. Ltd
1387 Zhangdong Road
Shanghai, China

ABSTRACT

In this paper, for efficient clustering of visual image data that have arbitrary mixture distributions, we propose a simple distance metric learning method called Maximum Normalized Spacing (MNS) which is a generalized principle based on Maximum Spacing [12] and Minimum Spanning Tree (MST). The proposed Normalized Spacing (NS) can be viewed as a kind of adaptive distance metric for contextual dissimilarity measure which takes into account the local distribution of the data vectors. Image clustering is a difficult task because there are multiple nonlinear manifolds embedded in the data space. Many of the existing clustering methods often fail to learn the whole structure of the multiple manifolds and they are usually not very effective. Combining both the internal and external statistics of clusters to capture the density structure of manifolds, MNS is capable of efficient and effective solving the clustering problem for the complex multi-manifold datasets in arbitrary metric spaces. We apply this MNS method into the practical problem of multi-view image clustering and obtain good results which are helpful for image browsing systems. Using the COIL-20 [19] and COIL-100 [18] multi-view image databases, our experimental results demonstrate the effectiveness of the proposed MNS clustering method and this clustering method is more efficient than the traditional clustering methods.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms, Similarity Measures*; E.1 [**Data Structures**]: Graphs and Networks

General Terms

Algorithms

Keywords

Data Clustering, Distance Metric Learning, Data Mining

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

1. INTRODUCTION

Clustering is the unsupervised learning of pattern grouping. It identifies groups of data, such that data in the same group are similar to each other, while data in different groups are dissimilar. Data clustering is an important kind of data analysis tasks. As a kind of data clustering, image clustering is a technique that associates each image in dataset with a class label such that the image associated with the same label are similar to each other. Image clustering is practically very useful for image management systems because digital image datasets are growing explosively in both number and size due to the rapid popularization of digital cameras and mobile phone cameras in the last decade. These large image collections require automatic clustering to facilitate browsing, manipulation and sharing of images with image management systems.

The problem of data clustering has been studied for decades [11, 28] and it is an active research field in machine learning and data mining. As a kind of data types considered in the clustering research, image is very difficult to be handled due to lack of understanding on their intrinsic properties. Usually, there are multiple nonlinear manifolds embedded in the image data space. So image clustering is a kind of multi-manifold data clustering. In Figure 1, there are some toy data points which have embedded multiple manifolds. According to the density distribution, three nonlinear manifolds can be identified on the data points in Figure 1. Image data are often shown to reside on such nonlinear embedding. Traditional clustering methods, such as K-means and Gaussian mixture model, often get poor results for multi-manifold data clustering. The reason why traditional methods failed is that the typical mixture of Gaussian distributions is defined on the Euclidean space, and hence it can not always describe the data points sampled from nonlinear manifolds.

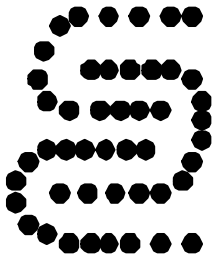

Figure 1: A case where some toy data points have multiple nonlinear manifolds.

Some feature extraction methods, such as ISOMAP [25], LLE [22] and semidefinite embedding [26], have been proposed for extracting the nonlinear manifold structure for data analysis and visualization. But it's difficult to directly apply these feature extraction methods to data clustering. A manifold clustering method [24] used MDS (multi-dimensional scaling) for data clustering. Another manifold clustering method [30] used ISOMAP for shape clustering. Currently the kernel clustering [9] and spectral clustering [23, 27, 20, 5, 32] methods have attracted increasing attention due to their promising performance for complex datasets. However, kernel and spectral clustering are still not good enough for image clustering. In our opinion, the reason why spectral clustering is not very good is that it is not specially adapted to multi-manifold data clustering. Self-tuning spectral clustering [32] used local scaling to improve clustering accuracy. Instead of selecting a single scaling parameter σ, local scaling parameter σ_i for each data point s_i was calculated in [32]. In [13], for spectral clustering, affinity function with multiple local scaling parameter σ has been proposed for improving the clustering performances. Spectral clustering was used for content-based image retrieval in [3]. Spectral clustering was used for hierarchical clustering of WWW image search results with visual, textual and link information in [1]. A method [16] used lossy data compression for segmenting data that have Gaussian mixture distributions. A method called affinity propagation [10] has been proposed for data clustering. Affinity propagation has some attractive properties but it is still not very good for multi-manifold data clustering because it can not capture the whole structure of nonlinear manifolds. A method with metric and local linear structure [15] has been proposed for image clustering and it obtained good results on COIL-20 [19] and COIL-100 [18] image databases. But this method [15] focused on computing-intensive distance metric learning and it is not an efficient clustering method. A global geometric method [33] with geodesic distance learning has been proposed for image clustering and it has an experimental result on COIL-20 database.

For the multi-manifold data clustering, we propose a clustering method using a simple new clustering criterion called Maximum Normalized Spacing (MNS) which is a generalized principle based on Maximum Spacing [12]. Combining both the internal and external statistics of clusters to capture the density structure of manifolds, MNS can effective and efficient solve the clustering problem for the complex multi-manifold datasets. With maximum normalized spacing, boundaries among manifolds can be identified and clustering is obtained by dividing the data points at the boundary of manifolds. So MNS is specially suitable for multi-manifold data clustering. MNS is different from the methods in [24, 30, 15, 33] because these methods mainly focused on computing-intensive distance metric learning with nonlinear dimensionality reduction and geodesic distance while MNS is a simple and efficient clustering method for arbitrary metric spaces with very simple distance metric learning which isn't computing-intensive. MNS method is applied into the practical problem of image clustering and good results which are helpful for image browsing systems are obtained. Our experimental results on the COIL-20 [19] and COIL-100 [18] image databases show that MNS method is consistently accurate, efficient and it has some advantages over some of the state-of-the-art clustering methods.

2. RELATED WORK

According to [34], there are internal, external and combining criterion functions to be optimized by clustering methods. The internal criterion functions focus on producing a clustering solution that optimizes a function defined only over the data within each cluster and does not take into account the data assigned to different clusters. The external criterion functions derive the clustering solution by focusing on optimizing a function that is based on how the various clusters are different from each other. The combining criterion functions focus on combining both internal and external characteristics of the clusters. The popular K-means algorithm uses an internal criterion function:

$$\text{minimize} \quad \mathcal{I}_1 = \sum_{r=1}^{k} \sum_{d_i \in S_r} D(d_i, O_r) \tag{1}$$

where k is the number of clusters, S_r is the dataset of the r-th cluster, d_i is the i-th data point, O_r is the centroid vectors of r-th cluster, and $D(d_i, d_j)$ is the distance between data points d_i and d_j. This internal criterion function does not take into account the differences among the data assigned to different clusters. The greedy nature of the K-means algorithm does not guarantee that it will converge to a global minima, and the local minima solution it obtains depends on the particular set of seed data points that were selected during the initial clustering. In order to improve clustering accuracy, researchers try to find some new clustering methods for real-world applications. The new clustering method affinity propagation [10] also has an internal criterion function:

$$\text{minimize} \quad E = -\sum_{i=1}^{N} s(i, c_i) \tag{2}$$

where c_i is the exemplar (representative data point) of the current cluster in which data point i lies, $s(i, c_i)$ is the similarity of data point i to its exemplar c_i and N is the number of data points of the dataset. $s(i, c_i) = -D(i, c_i)$ in [10]. The affinity propagation [10] is different from K-means because it tries to find clusters' exemplars instead of means and optimizes the internal criterion function (2) based on exemplars.

For meaningful clustering solutions, external criterion functions are hard to be defined [34]. Actually with an external criterion function, an old clustering method [31] used minimum spanning tree (MST) for data clustering. Using MST representation of the dataset, this MST clustering method [31] generates clusters by deleting the MST edges with the largest lengths. Gene expression data clustering [29] and image segmentation [8] were studied based on MST. Clustering of maximum spacing was defined in [12]. In [12], the "spacing" is defined as the minimum distance between any pair of data points in different clusters. A similar concept was also defined as difference between two components in [8]. Clustering of maximum spacing [12] can be produced using this MST clustering method [31].

3. MAXIMUM NORMALIZED SPACING FOR CLUSTERING

Combining both the internal and external statistics of clusters, maximum normalized spacing is a kind of partitional clustering methods. Partitional clustering and agglomerative clustering are two kinds of methods for hierarchical clustering. Contrary to the common belief, the experimental evaluation for document clustering in [34] shown that partitional clustering methods always lead to better solutions than agglomerative clustering methods. We propose maximum normalized spacing (MNS) for partitioning data points to form clusters. This clustering criterion is a generalized principle based on maximum spacing [12] and MST. For complex multi-manifold datasets that have arbitrary mixture distributions, the clustering can be obtained through maximum normalized spacing with respect to local data density distributions.

3.1 Looking for New Clustering Criterion

According to [12], spacing is the minimum distance between any pair of data points in different clusters. The MST clustering method [31] has the following criterion for maximizing spacing [12]

$$\text{maximize} \quad \mathcal{S} = \min_{d_i \in S_q, d_j \in S_r (q,r=1,2,\ldots,k; \quad q \neq r)} \{D(d_i, d_j)\} \tag{3}$$

where $\mathcal{S}$ denotes the spacing of a k-clustering problem (the k-clustering is to divide data points into k non-empty groups), S_q and S_r are two different clusters. $D(d_i, d_j)$ is the distance between data points d_i and d_j. This is an external criterion function. As a result after optimizing the criterion function (3), the generated k clusters have maximum spacing which is denoted as $SP(k)$

$$SP(k) = \max_{t=1,2,\ldots,T} \{\mathcal{S}_t\} \tag{4}$$

$$= \max_{t=1,2,\ldots,T} \{ \min_{d_i \in S_q^{(t)}, d_j \in S_r^{(t)} (q,r=1,2,\ldots,k; \quad q \neq r)} \{D(d_i, d_j)\}\} \tag{5}$$

where T is the number of all possible solutions for a k-clustering problem on a dataset and $\mathcal{S}_t$ is the spacing in the t-th clustering solution. $S_q^{(t)}$ and $S_r^{(t)}$ are two different clusters in the t-th clustering solution. On a dataset with N data points, the k clusters ($k = 2, 3, \ldots, N$) generated by the MST clustering method has maximum spacing $SP(k)$ which is uniquely determined by the MST of the dataset because $k-1$ largest MST edges have been deleted. Therefore, for various values of k (the number of clusters to be generated), there are $N-1$ candidate spacings

$$(SP(2), SP(3), \ldots, SP(N))$$

for a dataset with N data points. These $N-1$ candidate spacings $(SP(2), SP(3), \ldots, SP(N))$ can be identified by constructing MST of the dataset and each spacing $SP(k)$ is equivalent to the length of the $(k-1)$-th largest edge e_k of the MST

$$SP(k) = L(e_k) = D(d_u, d_v), \quad k = 2, 3, \ldots, N \tag{6}$$

where $L(e_k)$ denotes the length of the edge e_k of the MST, d_u and d_v are the two vertexes of the edge e_k. So the $N-1$ spacings $(SP(2), SP(3), \ldots, SP(N))$ are associated with the $N-1$ MST edges $(e_2, e_3, \ldots, e_N)$ one-to-one respectively through their relationships according to equation (6). For example, $SP(2)$ is associated with the edge e_2 which is the MST edge with the largest length. The edge set $\{e_k\}$ has the non-increasing order according to the edge lengths

$$L(e_{k-1}) \geq L(e_k), \quad k = 3, 4, \ldots, N \tag{7}$$

The set $\{SP(k)\}$ also has the non-increasing order

$$SP(k-1) \geq SP(k), \quad k = 3, 4, \ldots, N \tag{8}$$

The MST clustering method is not very robust for some complex datasets because its external criterion function (3) neglects the relationships among the data within each cluster. Simply breaking large MST edges is inadequacy and it is shown in Figure 2 where some toy data points have been partitioned. In Figure 2, the MST clustering method just produces the bad partitioning and the good partitioning is be neglected because the internal statistics of clusters can not be considered by the maximum spacing. A modification was made for this method in [6]. According to [6], inconsistent edges of MST should be deleted for generating clusters. In [6], an edge is inconsistent if its length is significantly larger than the average length of all other edges incident on its nodes. This modified method has improved the MST clustering method in some conditions. However, as discussed in [6], this modified method is still

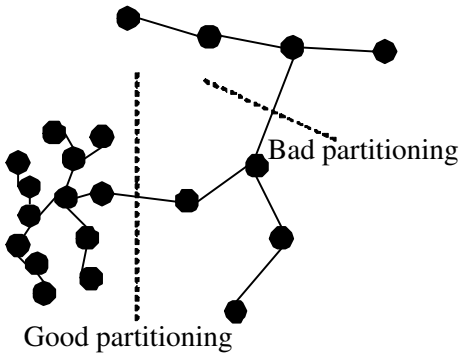

Figure 2: A case where some toy data points have been partitioned.

sensitive to local conditions and not accurate for some complex situations.

Combining both the internal and external views of the clustering process should be made for improving the clustering accuracy. Following this direction, we define normalized spacing $NSP(k)$ as

$$NSP(k) = \frac{SP(k)}{\max_{e_i \in C_{k,a}} \{SP(i)\}} + \frac{SP(k)}{\max_{e_j \in C_{k,b}} \{SP(j)\}} \tag{9}$$

where $C_{k,a}$ and $C_{k,b}$ ($k = 2, 3, \ldots, N$) are two connected components of MST and they are both connected with the edge e_k. d_a and d_b are two vertexes of the edge e_k. These two connected components are respectively constructed by greedily selecting M smallest edges through the sequential forward selection [21] from the two sides of the edge e_k. In the sequential forward selection [21], $C_{k,a}$ is growing from one side d_a of the edge e_k and $C_{k,b}$ is growing from another side d_b of the edge e_k. In this growing process, the edges which are directly connected to the connected component are sorted and the smallest one is selected into the connected component. For example, the first edge selected into $C_{k,a}$ is the smallest one among all the edges (except edge e_k) incident on vertex d_a. This growing step is repeated until M edges are selected. The following is the algorithm for constructing connected component $C_{k,a}$ ($C_{k,b}$ is constructed in the same way):

- Initialize: $C_{k,a} = \varnothing$, $E = \{e_i\}$ (E is initialized as the set of all the edges (except the edge e_k) which are direct incident on the vertex d_a
- For $t = 1, 2, \ldots, M$
  1. Select the minimum edge e^t in E (e^t has vertex d^t which is connected with the edges out of E and $d^t \neq d_a$) and move e^t out of E
  2. $C_{k,a} = C_{k,a} \cup e^t$
  3. Add all the edges (except the edge e^t) which are direct incident on the vertex d^t into E
- Output: $C_{k,a}$

$C_{k,a}$ and $C_{k,b}$ do not include the edge e_k but they both have at least one edge which is directly connected to the edge e_k. $C_{k,a}$ and $C_{k,b}$ are not connected with each other and they both have M edges:

$$M = Num(C_{k,a}) = Num(C_{k,b}), \quad M < N \tag{10}$$

where $Num(C_{k,a})$ denotes the number of edges in $C_{k,a}$ and N is the number of data points of the whole dataset. $C_{k,a}$ and $C_{k,b}$ represent two denser neighborhoods beside the edge e_k and they are used for measuring local density information. The connected component $C_{k,a}$ tries to optimize the following criterion function

$$\text{minimize} \quad U = \sum_{e_i \in C_{k,a}} L(e_i) \tag{11}$$

For equation (11), $C_{k,a}$ is not the exact solution and it's an approximate solution which is obtained through the sequential forward selection [21]. This problem is similar to the K-minimum spanning tree (K-MST) problem and the K-MST problem is known to be NP-complete.

As shown in equation (9), the normalized spacing $NSP(k)$ is computed as the sum of two fractions which are ratios of the spacing $SP(k)$ to the maximum spacing among its neighborhood spacings. The normalized spacing is relative to its neighborhood density and different from the original spacing [12] which is absolute on the whole. Therefore, both the internal and external statistics of clusters can be measured by normalized spacing which is adapted to multi-manifold data.

The normalized spacing is inspired by a previous work called min-max cut [5]. For k-clustering problem, min-max cut [5] is defined as

$$\text{maximize} \quad \sum_{r=1}^{k} \frac{\sum_{d_x \in S_r, d_y \in S-S_r} D(d_x, d_y)}{\sum_{d_i, d_j \in S_r} D(d_i, d_j)} \tag{12}$$

where S_r is the set of data in a cluster and $S - S_r$ is the set of the rest of data in the dataset. The style of min-max cut for combining both the internal and external views of the clustering process is similar to that of normalized spacing but normalized spacing is simpler and easier to be computed than min-max cut because normalized spacing is computed on MST instead of complete graph. The normalized cut [23] also used the similar combining style. Computed on complete graph, min-max cut problem is NP-complete because of its combinatory nature. Computed on MST, efficient greedy algorithm can be used for normalized spacing.

Simple new clustering criterion, maximum normalized spacing, is defined as follows

$$NSP(i) = \max_{k=2,3,\ldots,N} \{NSP(k)\} \tag{13}$$

This is a combining criterion function. This new criterion tries to maximize the external spacing while simultaneously minimize internal sparsity of clusters. Both the internal and external views of the clustering process are considered by this new criterion function. This new clustering criterion is a generalized principle based on maximum spacing [12]. There are $N-1$ normalized spacing $(NSP(2), NSP(3), \ldots, NSP(N))$ for a dataset with N data points. We can associate the $N-1$ normalized spacing

$$(NSP(2), NSP(3), \ldots, NSP(N))$$

with the $N-1$ MST edges $(e_2, e_3, \ldots, e_N)$ one-to-one respectively. Therefore, it can be seen that an edge e_k of MST has an associated normalized spacing $NSP(k)$. According to the criterion function (13), MNS clustering method generates clusters by deleting the MST edge e_i which is associated with the maximum normalized spacing $NSP(i)$. MNS method can be used to compute hierarchical clustering solutions using repeated cluster bisectioning. In MNS, all the data points are initially partitioned into two clusters with maximum normalized spacing. Then, one of these clusters containing more data points than others is selected and is further bisected according to the maximum normalized spacing. This process continues $k-1$ times and produces k clusters with maximum normalized spacing. In addition, being different from [12] in which Kruskal's algorithm was used, Prim's algorithm is used for constructing MST of the dataset for MNS because MNS is a graph-breaking method and not a graph-growing method for which Kruskal's algorithm is used.

3.2 Determining the Number of Clusters

For determining the number of clusters, the method via coding length [16] should be considered. In [16], the coding length function subject to the squared error ε^2 for encoding the m vectors in $W \subset \mathrm{R}^n$ from a Gaussian distribution is

$$L(W) \doteq \frac{m+n}{2} \log_2 \det \left(I + \frac{n}{m\varepsilon^2} WW^T \right) \tag{14}$$

For nonlinear manifold data that have arbitrary mixture distributions, we replace matrix WW^T with the kernel Gram matrix and modify the coding length function (14) as

$$L(W) \doteq \frac{m+n}{2} \log_2 \det \left(I + \frac{n}{m\varepsilon^2} \mathrm{K} \right) \tag{15}$$

where K is the kernel Gram matrix with element K_{ij}

$$\mathrm{K}_{ij} = k(w_i, w_j) \tag{16}$$

For a given distance metric $D(w_i, w_j)$, such as geodesic distance [25] or the χ^2 distance:

$$D(w_i, w_j) = \sum_x \frac{(w_{i,x} - w_{j,x})^2}{w_{i,x} + w_{j,x}} \tag{17}$$

kernel $k(w_i, w_j)$ can be computed as:

$$k(w_i, w_j) = \frac{1}{2} \left(D(w_i, w_o)^2 + D(w_j, w_o)^2 - D(w_i, w_j)^2 \right) \tag{18}$$

where w_o is the centroid vector which can be easily found. Centering matrix [25] also can be used to convert distances to inner products for Gram matrix. The descent of coding length [16] for bisectioning $S_1 \cup S_2$ into S_1 and S_2 is

$$H = L^s(S_1, S_2) - L^s(S_1 \cup S_2) \tag{19}$$

where

$$L^s(S_1, S_2, \ldots, S_k) \doteq \sum_{i=1}^{k} L(S_i) + |S_i|(-\log_2(|S_i|/m)) \tag{20}$$

For finishing the clustering, the cluster bisectioning step can be automatically stopped when $H \geq 0$ according to the equation (19) . For efficiency, this computing on equation (19) can only be made when the bisectioning steps are near finishing clustering because this computing is expensive.

3.3 Speeding Up for Large Databases

The main computationally expensive step in MNS is the computation of MST. It takes $O(N^2)$ time for constructing MST using Prim's algorithm for MNS. While some new algorithms have been proposed for constructing MST, such as Bernard Chazelle's soft heap [2], Prim's algorithm is still suitable for us because our problem is based on complete graph. For large databases, MNS is still computationally expensive. To speed up clustering process for large clustering problems, we use the canopies [17] as divide-and-conquer strategy for constructing MST for MNS. Using overlapping canopies for very large databases, we can construct approximate MST (AMST) for MNS through modular hierarchies and parallel computing. Large clustering problems that were formerly

impossible become practical through AMST and MNS. This reduction in computational cost comes without much loss in clustering accuracy.

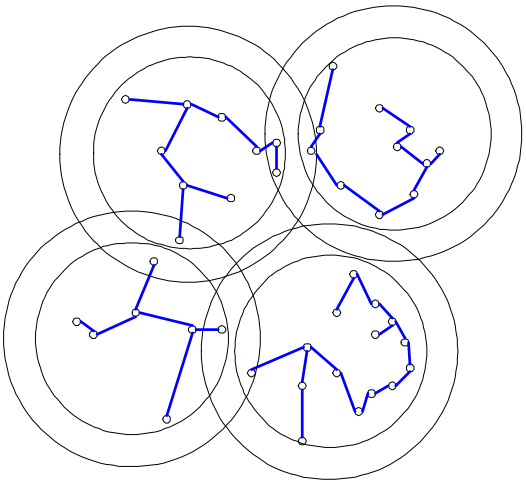

Figure 3: In four overlapping canopies, four MSTs are constructed respectively using Prim's algorithm on some toy data points.

AMST is constructed based on overlapping canopies. According to [17], start creating canopies with a list of the data points in any order, and with two distance thresholds, T_1 and T_2, where $T_1 > T_2$. Pick a data point randomly from the list and measure its distance to all other data. Put all data that are within distance threshold T_1 into a canopy. Remove from the list all data that are within distance threshold T_2. Repeat until the list is empty and canopies are created on the dataset. In each canopy, MST is constructed respectively using Prim's algorithm. At last, all MSTs in different canopies are merged to form a single large AMST. Different MSTs are connected by the smallest edges using Kruskal's algorithm based on overlapping data points between different canopies. For example, the overlapping data points between two canopies have to be connected by the smallest edge using Kruskal's algorithm to merge the two MSTs inside these two canopies. It's interesting that both Prim's and Kruskal's algorithms are used for constructing AMST. An example is shown in Figure 3 and four small MSTs are constructed respectively in four overlapping canopies using Prim's algorithm. In Figure 4, these four small MSTs are merged to form a single large AMST. The red edges have connected the four MSTs to form a single spanning tree using Kruskal's algorithms in Figure 4. AMST can be constructed by parallel computing and this computing can be based on the efficient systems of Google's MapReduce [4]. MNS can speed up for large databases based on the efficient computing of AMST through the parallel computing of MapReduce.

We use the AMST as data graph which is a data structure for data indexing and organization. we use the graph shrinking method to index the data points. As shown in Figure 5 and Figure 6, the end points of the data graph have been found and removed from the graph. This removing action is called graph shrinking. For the shrink graph, some former normal points become end points and these new end points can also be removed. So the graph shrinking process can be repeated many times and all the points in the data graph can be removed. As a result, all the data points can be ranked based on the removing sequence. So hierarchical data structure can

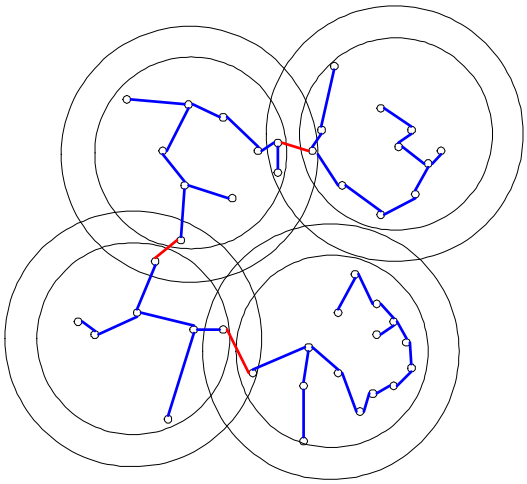

Figure 4: Four MSTs are connected to form a single large approximate MST (AMST) using Kruskal's algorithm.

be constructed in this way. This hierarchical data structure makes the clustering process very efficient.

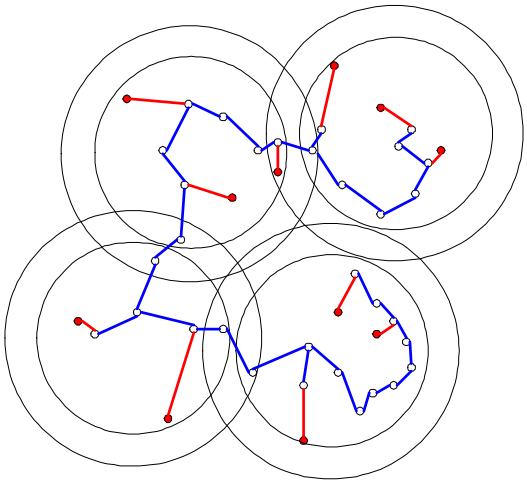

Figure 5: The end points (red points) of the data graph have been found.

4. EXPERIMENTS

In this section, we report our experimental results. To evaluate the performance of MNS, we compare MNS with some state-of-the-art clustering methods. We use the COIL multi-view image databases of Columbia which are popular databases for 3D object recognition problems. There are two COIL databases, COIL-20 [19] and COIL-100 [18]. They contain 20 and 100 objects, respectively. For both databases, the images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. There are 1440 images in COIL-20 and 7200 images in COIL-100. In Figure 7, some example images of 20 objects in COIL-20 database are shown. In Figure 8, some example multi-view images of a object in COIL-20 database are shown. In Figure 9, some example images of 100 objects in COIL-100 database are shown. We sub-sampled the image collections because the sampling in COIL databases is very dense. We let COIL20.2 denote

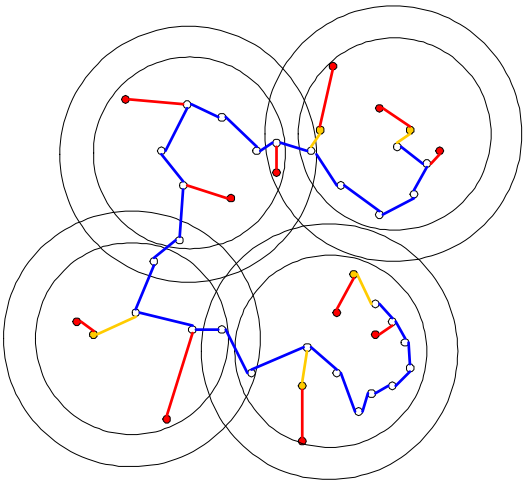

Figure 6: After graph shrinking, the new end points (yellow points) of the data graph have been found.

the collection of images obtained from COIL-20 by sub-sampling it with a factor of 2. So COIL20.2 contains the same number of objects as the original COIL-20 but with half as many images per object. Similarly, COIL20.4 denotes the collection obtained from COIL-20 by sub-sampling it with a factor of 4 and so on. All experiments were made on PC of Intel Core Duo 2 GHz CPU with 2 GB RAM.

Figure 7: Example images of 20 objects in COIL-20 database.

We use an image texture feature called LPCIH (local patterns constrained image histograms) [7] as the image feature for image clustering in our experiments. As in [7], we use the following image distance metric

$$D(d_x, d_y) = \sum_i \frac{|d_{x,i} - d_{y,i}|}{d_{x,i} + d_{y,i}} \tag{21}$$

This distance metric is similar to the distance of Chi square statistic (χ^2).

We use clustering accuracy to measure clustering performance as follows:

$$AC = \frac{\sum_{i=1}^{N} \delta(y_i, map(r_i))}{N} \tag{22}$$

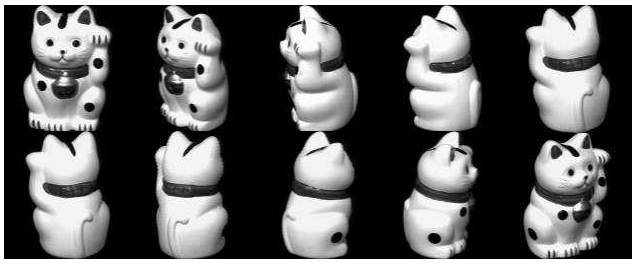

Figure 8: Some example multi-view images of a object in COIL-20 database.

Figure 9: Some example images of 100 objects in COIL-100 database.

where N is the number of data points in the dataset, $\delta(u, v)$ is the delta function that equals one if $u = v$ and otherwise equals zero, y_i and r_i are the groundtruch label and obtained cluster label respectively, and $map(r_i)$ is a function to map the cluster label to the ground truth label. In our experiments, the map function $map(r_i)$ is chosen to map the cluster label to the majority groundtruth label of the each estimated cluster.

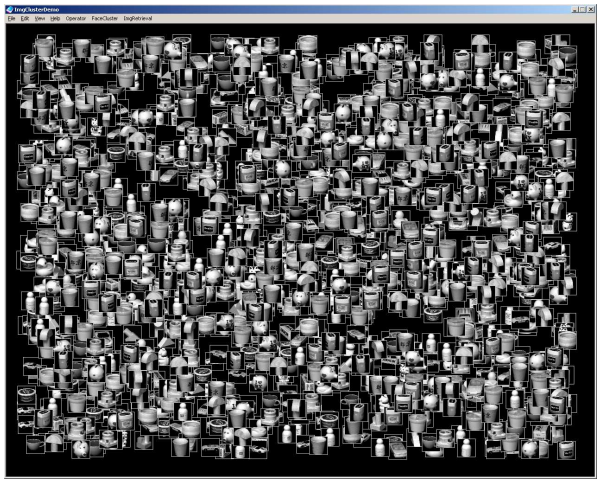

Figure 10: Our demo software loads the image set from COIL-20 database.

On COIL-20 and COIL-100 databases, the results of clustering accuracies of our algorithm and other three existing algorithms are reported in Table 1. We compare our algorithm with the methods in [15] and in [33]. In Table 1, the MST method denotes the MST clustering method [31]. We cite Lim's experimental results in [15] and Zhang's experimental results in [33] for comparing in Table 1. According the experimental results shown in Table 1, we can

Table 1: Results of clustering accuracies of our algorithm and other three algorithms.

Databases	COIL20	COIL20.2	COIL20.4	COIL20.8	COIL100	COIL100.2	COIL100.4
MNS	99.31%	95.69%	87.50%	78.89%	86.22%	79.31%	72.28%
Lim's results [15]	100%	94.86%	80.56%	-	-	79.31%	65.11%
MST	96.39%	91.25%	85.28%	74.44%	83.18%	76.25%	67.00%
Zhang's results [33]	-	93.80%	-	-	-	-	-

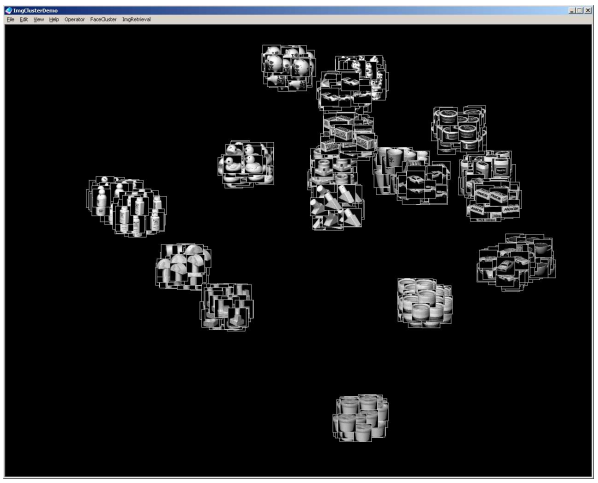

Figure 11: Our demo software shows the visual clustering result for COIL-20 database.

see that MNS algorithm is more accurate comparing with the other three existing methods. In Figure 10 and Figure 11, our demo software shows the visual results of image clustering based on COIL-20 database.

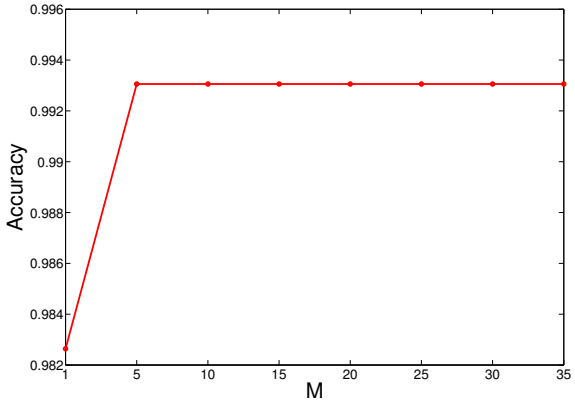

Figure 12: Clustering accuracy on COIL-20 database using MNS method.

In Figure 12, clustering accuracy on COIL-20 database using MNS method is shown. In Figure 12 , the horizontal coordinate M denotes the $Num(C_{k,a})$ in equation 10. In Figure 13, clustering accuracy on COIL20.2 database using MNS method is shown. From the results shown in Figure 12 and 13 , we can see that the performances of MNS are robust with respect to the changes of the parameter M.

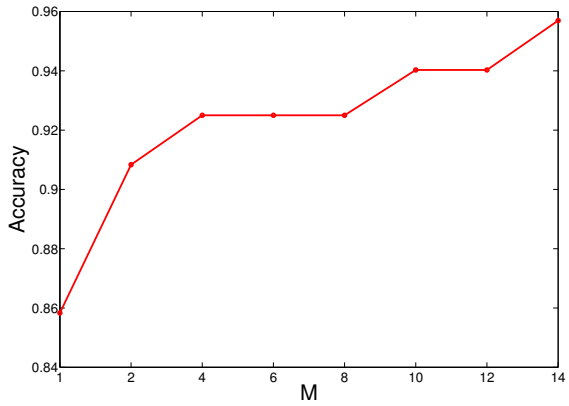

Figure 13: Clustering accuracy on COIL20.2 database using MNS method.

In Figure 14, clustering accuracies on COIL-20 database using MNS method and other three existing methods are shown. In Figure 15, CPU times of the clustering on COIL-20 database using MNS method and other three existing methods are shown. In Figure 14 and 15, the horizontal coordinate K denotes the number of the clusters,"SC" denotes the spectral clustering [23],"AP" denotes the affinity propagation [10], and "KM" denotes K-means.

In Figure 16, clustering accuracies on COIL20.2 database using MNS method and other three existing methods are shown. In Figure 17, CPU times of the clustering on COIL20.2 database using MNS method and other three existing methods are shown. In Figure 14, 15, 16, and 17, we can see that the performances of MNS are robust comparing with the three existing methods (spectral clustering, affinity propagation and K-means).

In Figure 18, clustering results produced by MNS on a small subset of the COREL [14] database are shown. These clustering results are used by our demo system of image browsing. We can see that such image clustering is helpful for browsing a large amount of images in photo albums and photo databases.

5. CONCLUSIONS

We propose a simple new clustering criterion called Maximum Normalized Spacing (MNS). This new clustering criterion is a generalized principle based on Maximum Spacing [12] for clustering. Combining both the internal and external statistics of clusters, MNS method is effective and efficient for the multi-manifold data clustering. MNS can speed up for large databases using divide-and-conquer strategy. MNS is different from the methods in [24, 30, 15, 33] because these methods mainly focused on distance metric learning with nonlinear dimensionality reduction and geodesic distance while MNS is a direct and simple clustering method for arbitrary metric spaces without distance metric learning. So MNS

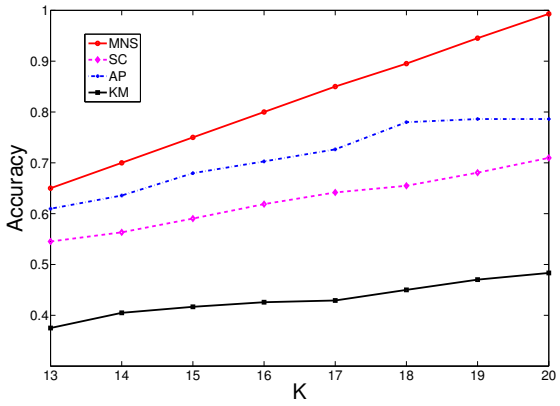

Figure 14: Clustering accuracies on COIL-20 database using MNS method and other three existing methods.

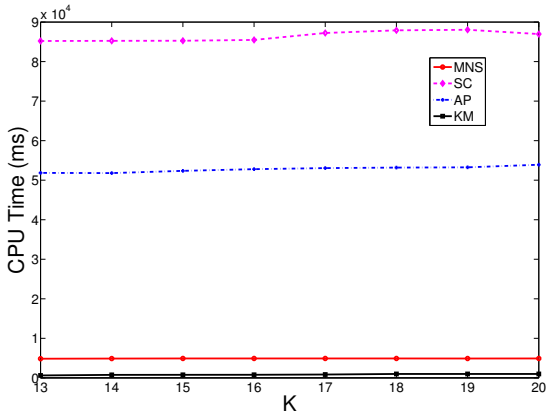

Figure 15: CPU times of the clustering on COIL-20 database using MNS method and other three existing methods.

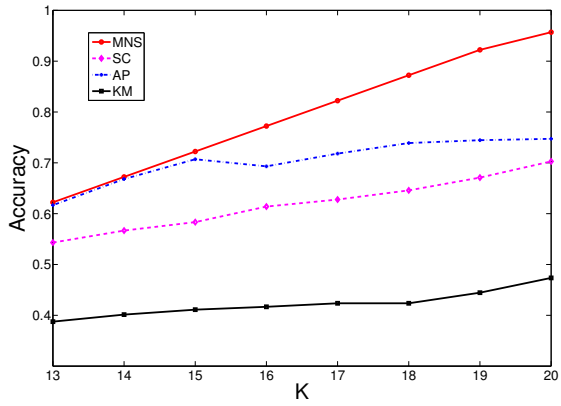

Figure 16: Clustering accuracies on COIL20.2 database using MNS method and other three existing methods.

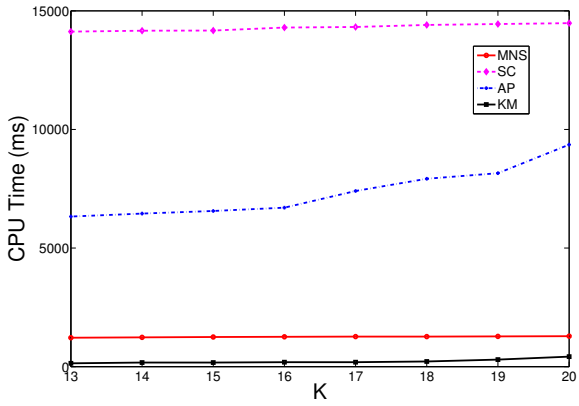

Figure 17: CPU times of the clustering on COIL20.2 database using MNS method and other three existing methods.

is convenient to use for real-world applications. MNS method is applied into the practical problem of image clustering in this paper and good results which are helpful for our demo system of image browsing are obtained. MNS can be used for many fields of real-world. For example, MNS is useful for image retrieval and image sharing on the social networking websites. Our experimental results show that MNS method is consistently accurate, efficient and it has some advantages over some of the state-of-the-art clustering methods.

6. REFERENCES

[1] D. Cai, X. He, Z. Li, W. Y. Ma, and J. R. Wen. Hierarchical clustering of www image search results using visual, textual and link information. *ACM Multimedia 2004*, pages 952–959, 2004.

[2] B. Chazelle. A minimum spanning tree algorithm with inverse-ackermann type complexity. *Journal of the ACM*, 47:1028–1047, 2000.

[3] Y. Chen, J. Z. Wang, and R. Krovetz. Clue: cluster-based retrieval of images by unsupervised learning. *IEEE Transactions on Image Processing*, 14:1187–1201, 2005.

[4] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *OSDI 2004*, pages 137–150, 2004.

[5] C. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. *ICDM 2001*, pages 107–114, 2001.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons Inc., 2nd edition, 2001.

[7] Z. G. Fan, J. Li, B. Wu, and Y. Wu. Local patterns constrained image histograms for image retrieval. *ICIP 2008*, pages 941–944, 2008.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[9] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41:176–190, 2008.

[10] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[11] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:264–323, 1999.

[12] J. Kleinberg and E. Tardos. *Algorithm design*. Addison Wesley, 1nd edition, 2005.

[13] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. *CVPR 2006*, 1:61–68, 2006.

[14] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1075–1088, 2003.

[15] J. Lim, J. Ho, M. Yang, K. Lee, and D. Kriegman. Image clustering with metric, local linear structure and affine symmetry. *ECCV 2004*, 1:456–468, 2004.

[16] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1546–1562, 2007.

[17] A. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. *KDD 2000*, pages 169–178, 2000.

[18] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). *Technical Report CUCS-006-96*, 1996.

[19] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-20). *Technical Report CUCS-005-96*, 1996.

[20] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. *NIPS 2001*, pages 849–856, 2001.

[21] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

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

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[24] R. Souvenir and R. Pless. Manifold clustering. *ICCV 2005*, 1:648–653, 2005.

[25] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[26] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70:77–90, 2006.

[27] Y. Weiss. Segmentation using eigenvectors: a unifying view. *ICCV 1999*, 1:975–982, 1999.

[28] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16:645–678, 2005.

[29] Y. Xu, V. Olman, and D. Xu. Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. *Bioinformatics*, 18:536–545, 2002.

[30] D. Yankov and E. Keogh. Manifold clustering of shapes. *ICDM 2006*, 1:1167–1171, 2006.

[31] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 20:68–86, 1971.

[32] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. *NIPS 2004*, pages 1601–1608, 2004.

[33] S. Zhang, C. Shi, Z. Zhang, and Z. Shi. A global geometric approach for image clustering. *ICPR 2006*, 4:960–963, 2006.

[34] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. *CIKM 2002*, pages 515–524, 2002.

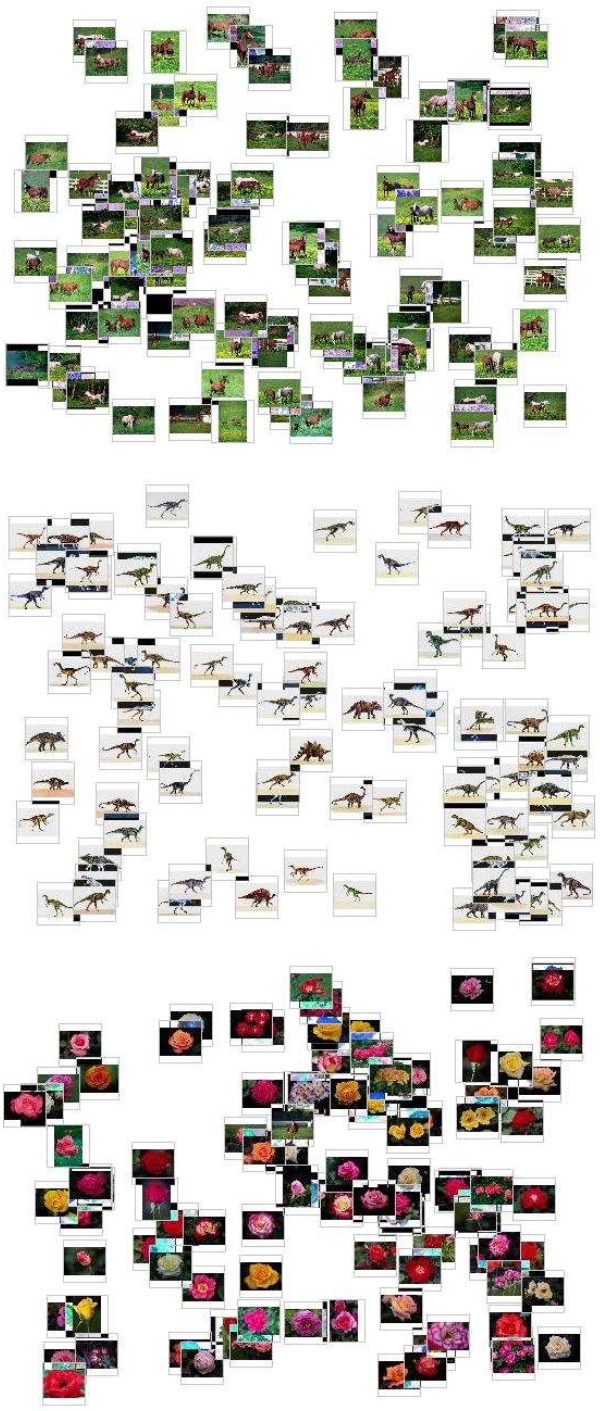

Figure 18: Clustering results produced by MNS on a small subset of the COREL [14] image database are used in our demo system.