Locality Condensation: A New Dimensionality Reduction Method for Image Retrieval

Zi Huang[†] Heng Tao Shen[†] Jie Shao[†] Stefan Rüger[§] Xiaofang Zhou[†]
[†]School of Information Technology and Electrical Engineering, The University of Queensland, Australia
[§]Knowledge Media institute, The Open University, United Kingdom
{huang,shenht,jshao,zxf}@itee.uq.edu.au s.rueger@open.ac.uk

ABSTRACT

Content-based image similarity search plays a key role in multimedia retrieval. Each image is usually represented as a point in a high-dimensional feature space. The key challenge of searching similar images from a large database is the high computational overhead due to the "curse of dimensionality". Reducing the dimensionality is an important means to tackle the problem. In this paper, we study dimensionality reduction for top-k image retrieval. Intuitively, an effective dimensionality reduction method should not only preserve the close locations of similar images (or points), but also separate those dissimilar ones far apart in the reduced subspace. Existing dimensionality reduction methods mainly focused on the former. We propose a novel idea called Locality Condensation (LC) to not only preserve localities determined by neighborhood information and their global similarity relationship, but also ensure that different localities will not invade each other in the low-dimensional subspace. To generate non-overlapping localities in the subspace, LC first performs an *elliptical condensation*, which condenses each locality with an elliptical shape into a more compact hypersphere to enlarge the margins among different localities and estimate the projection in the subspace for overlap analysis. Through a convex optimization, LC further performs a *scaling condensation* on the obtained hyperspheres based on their projections in the subspace with minimal condensation degrees. By condensing the localities effectively, the potential overlaps among different localities in the low-dimensional subspace are prevented. Consequently, for similarity search in the subspace, the number of false hits (i.e., distant points that are falsely retrieved) will be reduced. Extensive experimental comparisons with existing methods demonstrate the superiority of our proposal.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada.

General Terms

Algorithms, Measurement, Experimentation

Keywords

Dimensionality Reduction, Top-k Image Retrieval, Locality Condensation

1. INTRODUCTION

Multimedia similarity search aims at automatically retrieving multimedia objects similar to query objects from large databases based on their visual content, usually represented by some high-dimensional low-level features, such as color, texture and shape [21, 3]. In particular, top-k retrieval (or k-nearest neighbor search) finds the k most similar objects with respect to a query [24, 15, 6, 21, 3]. Since the dimensionality of a feature space is usually very high (up to hundreds of dimensions), directly indexing the original high-dimensional feature space usually fails for effective search due to the known phenomenon of the "curse of dimensionality" [27]. Dimensionality reduction that maps the original data onto a low-dimensional subspace becomes a promising way to alleviate this problem. Generally, dimensionality reduction can be used for the following purposes:

Simplifying complex data: For many applications, particularly in database and information retrieval, a high dimensionality of the feature space leads to high complexity of the data representation. The dimensionality has to be reduced to achieve a fast query response for an indexing structure or retrieval method. Typical methods include Discrete Fourier transform (DFT) and Discrete Wavelet Transform (DWT) [28], Adaptive Piecewise Constant Approximation (APCA) [7], Principal Component Analysis (PCA) [20] and its various improvements [26, 8, 24], Latent Semantic Indexing (LSI) [10] and its variants [17], Locality Preserving Projection (LPP) [16, 32, 14, 5], etc. For these types of applications, the process of dimensionality reduction must have an explicit mapping function to map the query points onto the low-dimensional subspace for similarity search.

Modelling and analyzing data: For many applications, particularly in machine learning and pattern recognition, the underlying data structure is often embedded in a much lower-dimensional subspace. The task of recovering meaningful low-dimensional structures hidden in high-dimensional observation data is also known as "manifold learning". Typical methods include Independent Component Analysis (ICA) [18], Multidimensional Scaling (MDS) [30], Isometric feature mapping (Isomap) [25] and its im-

provement [9], Locally Linear Embedding (LLE) [23], Laplacian Eigenmaps (LE) [1], Semantic Subspace Projection (SSP) [31], Maximum Margin Projection (MMP) [13], etc. For these types of applications, the reduction of dimensionality is typically a very expensive process and performed on a small set of sample points for learning purposes. Since they are defined only on the sample/training data and have no explicit mapping function, they are not directly applicable to information retrieval and database applications. Nonetheless, some recent proposals, such as Spectral Regression (SR) [6] and Laplacian Optimal Design (LOD) [15], have been utilized in relevance feedback for image retrieval.

In this paper, we are interested in dimensionality reduction for simplifying high-dimensional data for the purpose of efficient multimedia similarity search with a particular focus on top-k image retrieval. Given a set of points (e.g., images) $X = \{X_1, \ldots, X_n\} \subset \mathbb{R}^D$, it is expected to derive a suitable low-dimensional subspace from X to produce a compact low-dimensional representation, denoted as $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ ($d < D$). Therefore, the original data points are mapped onto a lower dimensional subspace, in which an effective indexing structure can be deployed, together with a fast search strategy (e.g., the framework presented in [22]). The similarity search will be performed in the reduced low-dimensional subspace. For measuring the quality of a dimensionality reduction method, two sets of search results (e.g., top-k results) returned from the reduced subspace and original space respectively need to be compared. *Precision*, defined as the overlap between two result sets divided by k, can indicate the reduction quality [24, 8]. Note that in different application domains, the precision can be defined differently.

Dimensionality reduction is a coin of two sides, both of which need to be considered for reduction quality: (1) nearby points in the original space should remain nearby in the reduced subspace, and (2) distant points in the original space should remain distant in the reduced subspace.

The widely used PCA [20] performs dimensionality reduction by projecting the original data points onto a linear low-dimensional subspace spanned by the top ranked principal components of the data. Its goal is to find a set of mutually orthogonal basis vectors that capture the directions of maximal variances in the original space while the reconstruction error is minimized. By doing so, distant points try to remain distant. However, the original space (e.g., image space) may not be linearly embedded, and in this case a problem arises. Figure 1 shows a sample of data points in a 2-dimensional space, where the data points are well separated into three partitions. By applying PCA, the original points are mapped onto a 1-dimensional subspace as shown in Figure 2. While the partitions marked with $*$ and $\bullet$ can be well distinguished from each other, both of them are heavily overlapped with another partition marked with $+$. Given a query point in the $+$ partition, the search results in the reduced 1-dimensional subspace are likely to contain originally distant data points from the $*$ or $\bullet$ partition. Therefore, directly applying PCA on non-linearly embedded data space may cause an unsatisfactory quality of the similarity search.

As an alternative to PCA, LPP has been proposed and applied in various database and information retrieval applications [16, 32, 14, 5]. LPP aims at preserving the locality of a neighborhood relationship and is claimed to be more useful in information retrieval applications. Given a set of points $\{X_1, \ldots, X_n\} \subset \mathbb{R}^D$, LPP achieves the optimal projections

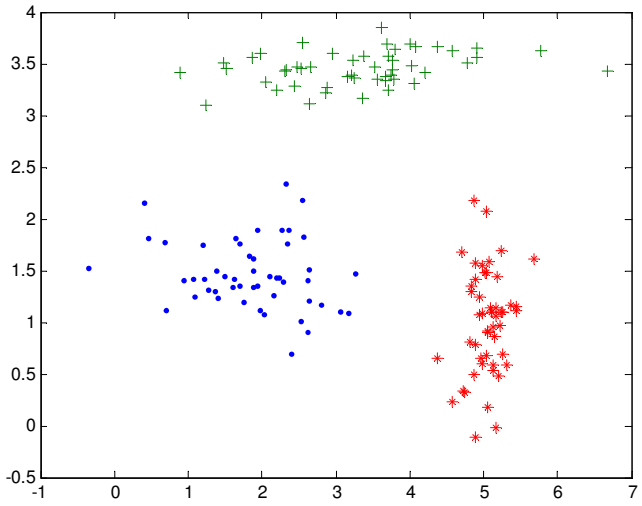


Figure 1: A sample set of 2-dimensional points

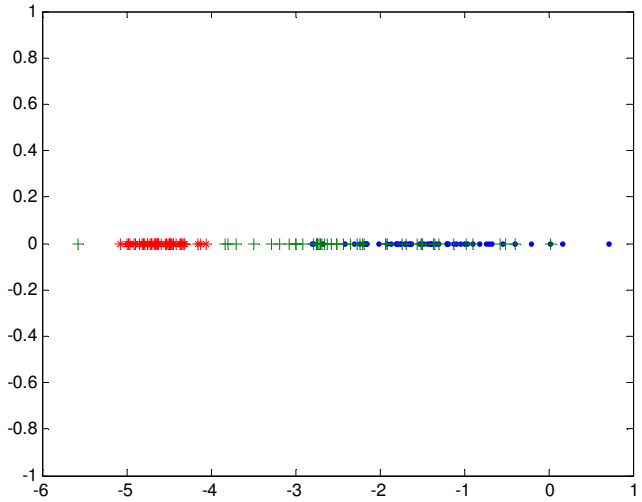


Figure 2: Dimensionality reduction by PCA

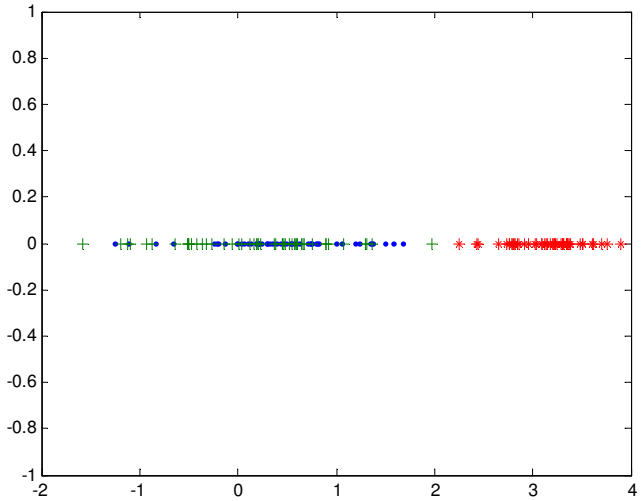


Figure 3: Dimensionality reduction by LPP

by minimizing the following function [16]:

$$\sum_{i,j} \text{dist}(x_i, x_j) \cdot W_{ij},$$

where dist() is the distance function between two projections x_i and x_j in the low-dimensional subspace, and W_{ij} evaluates the local structure of the original space (i.e., the similarity between X_i and X_j in $\mathbb{R}^D$). The above objective function attempts to ensure that nearby points remain nearby. Since LPP is designed for preserving the local structure, it is likely that a similarity search in the low-dimensional subspace returns similar results to that in the original space. In the objective function of LPP, a heavy penalty applies if nearby points X_i and X_j are mapped far apart in the subspace. However, it also leads distant points to be mapped nearby for a smaller $\text{dist}(x_i, x_j)$. Although LPP preserves locality, it is incapable of keeping distant points distant. Figure 3 shows the 1-dimensional projections of the sample data points in Figure 1 by applying LPP. Compared with the case of PCA in Figure 2, it is evident that the $*$ partition is better preserved by LPP since the $*$ partition in Figure 2 is much denser than that in Figure 3. However, the overlap between the other two partitions in Figure 3 is much heavier than that in Figure 2. This suggests that in LPP, different localities may invade each other. This potentially deteriorate the quality of dimensionality reduction for similarity search.

This paper proposes a new global dimensionality reduction method called Locality Condensation (LC), which is particularly designed for multimedia similarity search involving high-dimensional feature databases. Unlike PCA or LPP, LC aims at addressing both sides of dimensionality reduction (i.e., nearby points remaining nearby and distant points remaining distant). Our underlying idea is that if in the lower dimensional subspace, localities and their global similarity relationship can be preserved while different localities can be prevented from invading each other, then search in the subspace is likely to yield more accurate results by excluding distant points (i.e., false hits) to the queries.

LC first discovers a given number of neighborhood localities based on clustering strategies that minimize intra-locality variance. It then condenses each locality by an *elliptical condensation* followed by a *scaling condensation*. In real data sets, each locality may exhibit certain degree of orientation. By taking the dimension variabilities into account, elliptical condensation condenses each locality with an intrinsic elliptical distribution (i.e., a hyperellipsoid) into a smaller hypersphere by the Mahalanobis distance. Therefore, the overlaps among different localities can be reduced and the projection of each locality in the subspace can be determined. To further eliminate any potential overlap among different localities in the subspace, scaling condensation is performed on the obtained hyperspheres when necessary. Given that the projection of a hypersphere in any subspace is a lower-dimensional hypersphere with the same radius, the overlaps among different hyperspheres in the subspace can be analyzed. Through a convex optimization, the degrees of scaling condensation can be minimized so that the hyperspheres can be preserved at maximal degrees, with the constraint that different hyperspheres do not invade each other in the subspace. Finally, all condensed localities are mapped into a low-dimensional subspace by PCA. In this way, data points within the same hypersphere maintain their intrinsic locality neighborhood relationship, while different localities are no longer overlapped in the subspace. This results in a more accurate search result, since while neighboring points are condensed and preserved, distant points are prevented from being projected into neighboring positions in the subspace. We verify the effectiveness of LC algorithm through comprehensive experiments on various image databases. Our experiments show that LC largely outperforms the previous dimensionality reduction methods including PCA and LPP in terms of precision of the tok-k search results.

The rest of this paper is organized as follows. Section 2 reviews some related work of dimensionality reduction. Our LC algorithm with its illustrations and discussions is provided in Section 3 followed by its justifications in Section 4. Experimental results are shown in Section 5. Finally, we provide concluding remarks and a plan for future work in Section 6.

2. RELATED WORK

A number of dimensionality reduction methods have been proposed for various applications in information retrieval, databases, machine learning, pattern recognition, etc. They can be broadly classified into two categories: linear and nonlinear methods.

Linear dimensionality reduction is designed to operate when the subspace is embedded almost linearly in the original space. The classical PCA [20] and MDS [30] are simple to implement, efficiently computable and guaranteed to discover the true structure of data lying on or near a linear subspace of the high-dimensional input space. PCA finds a low-dimensional embedding of the data points that best preserves their variance as measured in the input space [20]. It identifies the directions that best preserve the associated variances of data points while minimizing the "least-squares" (Euclidean) error measured by analyzing the covariance matrix. PCA makes a stringent assumption of orthogonality. MDS finds an embedding that best preserves the inter-point distances by analyzing Gram matrix [30]. Both PCA and MDS choose a transformation matrix to project the original data set onto a lower dimensional subspace. However, many data sets contain intrinsically nonlinear structures that are invisible to PCA or MDS. For example, both methods fail to well separate the three clusters, shown in Figure 1, in the reduced 1-dimensional subspace. Meanwhile, MDS has a drawback that it has no explicit mapping function. If a new object (or query) comes, we do not know where it should be mapped to. Some local dimensionality reduction methods based on PCA such as [8, 24] are nonlinear. However, they are not designed to represent the global structure of a data set within a single coordinate system. As a result, the global structure cannot be recovered and visualized. Some dimensionality reduction methods have also been proposed particularly for time series databases [28, 7].

Other nonlinear dimensionality reduction methods are normally used for "manifold learning". Given the neighborhood information of each data point, nonlinear dimensionality reduction can reflect the intrinsic geometric structures of the underlying manifold well. Global approaches, such as Isomap [25], attempt to preserve the global geometry by geodesic distances instead of Euclidean distances. Isomap computes the local distances of neighboring data points and uses the shortest paths connecting them as the estimated geodesic distances for faraway points to learn the underlying

global geometry, and then MDS is applied to the matrix of graph distances to find a set of low-dimensional points that attempts to preserve the geodesic manifold distances among all pairs of data points. Local approaches, such as LLE [23] and Laplacian Eigenmaps [1], attempt to preserve the local geometry of data and recover global nonlinear structure from locally linear fits. LLE adopts a local flavor without the need of estimating pairwise distances between the widely separated data points. It assumes each data point with its neighbors lies on or close to a locally linear patch of the manifold and recovers the global nonlinear structure from the local geometry of these patches. The coefficients of the best approximation are a weighted linear combination of its neighbors. Laplacian eigenmap [1] also preserves local neighborhood information. The Laplace-Beltrami operator is approximated by the weighted Laplacian of the adjacency graph with the weights chosen appropriately. However, similar to MDS, these nonlinear methods are defined on the training data set only and cannot be directly applied for similarity search.

Based on Laplacian Eigenmaps, LPP [16] is an optimal linear approximation to the eigenfunctions of the Laplace-Beltrami operator on the manifold. Using the notion of the Laplacian of the graph, a transformation matrix, which minimizes the sum of products of inter-distances between any two neighboring points in the low-dimensional subspace and original space, is computed and used to map the data points onto the subspace. LPP has been applied with success in various information retrieval applications to preserve the local neighborhood structures of the data sets [16, 32, 14, 5].

Locality based mapping [16] preserves a large amount of proximity structure information of data, meaning that nearby points tend to get mapped to be nearby. *While maintaining the original neighborhood relationship within each locality is important, here we emphasize that it is also crucial to avoid any potential overlap among different localities in the reduced subspace after dimensionality reduction.* Our Locality Condensation algorithm, which is a global dimensionality reduction method based on the classical PCA, aims at addressing this problem. It inherits the locality preserving quality and further exploits the other side of the coin. By condensing localities to keep distant localities separate, our goal is to achieve a better discriminative capacity. This is particular useful for top-k retrieval (or k-Nearest neighbor search) because the number of false hits (i.e., distant points to a query that are falsely retrieved) is expected to be reduced. Similar to LPP, LC is also defined everywhere in the input space, rather thnan just on the training data points (which may cause an "out-of-sample" problem [2]). This property not only remedies the crucial issue of dynamic update in large databases but also implies that we may use a smaller number of sample points and then similarly map all the points in the database onto a subspace for facilitating similarity search.

3. LOCALITY CONDENSATION

Locality Condensation (LC) considers not only preserving each individual locality but also separating different localities in the subspace. For the convenience of presentation, we first describe the basic LC algorithm, then provide the illustrations for each step in detail, followed by further discussions. Table 1 provides a list of notations used.

Symbols	*Descriptions*
D	Dimensionality of the original space
d	Dimensionality of the subspace
X_i	A point in the original space
x_i	X_i's projection in the subspace
G_i	i^{th} locality
m	Number of localities
O_i	Center of G_i in the original space
o_i	O_i's projection in the subspace
R_i	Radius of G_i after elliptical condensation
r_i	Radius of G_i after scaling condensation
M_i	Inverse of covariance matrix for G_i
Φ_d	Matrix containing 1^{st} to d^{th} PCs
k	Number of nearest neighbors

Table 1: Table of notations

3.1 The Algorithm

As a whole, the proposed LC algorithm firstly seeks to generate localities based on the neighborhood relationship among data points, then condenses the localities when necessary to enlarge the margins of different localities with the guarantee that the space of each locality is dissociated from one another in the subspace while each locality can be maximally preserved, and finally projects the condensed localities onto the subspace. Our algorithm consists of three main steps as stated below:

1. **Generate Localities**

   In this step, multiple localities are obtained by partitioning the data points into a number of clusters based on their neighborhood relationship. Given a set of points $X = \{X_1, \ldots, X_n\}$ in $\mathbb{R}^D$ space, a partitioning strategy that minimizes an objective function across different clusters can be applied to generate a set of localities, denoted as $G = \{G_1, G_2, \ldots, G_m\}$. The most common method for data clustering is k-means algorithm, which minimizes the overall intra-cluster variance. Convergence is achieved when the points no longer switch clusters. In this paper, we use the fast version of standard k-means algorithm [11] to generate m clusters, each of which represents a locality composed of neighboring points.

2. **Condense Localities**

   This step consists of elliptical condensation and scaling condensation.

   2.1: Elliptical Condensation

   A locality G_i can be modelled as a hypersphere centered at the mean of data points with the radius of maximal distance of points to the center. The reason for representing a locality as a hypersphere is that a hypersphere's projection in any subspace can be easily estimated for overlap analysis. More details will be discussed later.

   However, real data sets often exhibit a certain degree of orientation [8, 24, 16]. For a locality G_i, its data distribution is more like a hyperellipsoid. Therefore, a hypersphere with a large radius may potentially "overfit" a locality. As shown in Figure 4, a locality G_i (the shaded area) can be modelled as a hypersphere (the

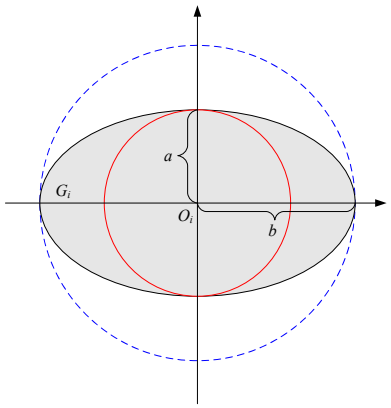


Figure 4: Locality modelling

outer dashed circle) with the radius b, which is also the radius of the hyperellipsoid containing G_i along its largest axis. Clearly, such a large hypersphere over-fits G_i greatly and may potentially cause heavy apparent overlaps with other localities.

To avoid the above problem, we can present the relationship among the points by the Mahalanobis distance, which assigns all points on a hyperellipsoid the same distance to the center. A hyperellipsoid can then be transformed into a compact and smaller hypersphere. The Mahalanobis distances among the points in G_i can be preserved as the Euclidean distances after a certain transformation, which is formally defined as follows:

First, by Singular Value Decomposition (SVD), M_i, which is the inverse of covariance matrix of G_i, can be decomposed into a product of three matrices:

$$M_i = U_i S_i V_i^T.$$

Second, X_i is mapped into Mahalanobis space by:

$$X_i^{\mathrm{m}} = (X_i - O_i) U_i \sqrt{S_i},$$

where O_i is the cluster center.

Third, X_i^{m} is transformed back into the original coordinate system as follows:

$$X_i' = X_i^{\mathrm{m}} U_i^{-1} + O_i.$$

By the above mapping, a locality with an elliptical data point distribution can be transformed into a hypersphere with its radius denoted as R_i'. The justification is shown in Section 4.1.

However, theoretically there is no clear scaling relationship between the Mahalanobis distance and the Euclidean distance. That is, the obtained hypersphere based on the Mahalanobis distance could have an even larger radius than the hypersphere containing G_i in the Euclidean space. Therefore, the following step is necessary to guarantee a compact hypersphere with a smaller radius:

$$X_i' \leftarrow (X_i' - O_i)\frac{R_i}{R_i'} + O_i,$$

where R_i is set to be *the radius of the hyperellipsoid containing G_i along its second largest axis.*

This process is called *elliptical condensation*, based on the Mahalanobis distance. After elliptical condensation, a locality G_i is modelled as a D-dimensional hypersphere with a radius of R_i, which can be denoted as R_i-hypersphere^D. Recall the example in Figure 4; by elliptical condensation, G_i is modelled as a smaller and compact hypersphere (the inner circle with the radius a) instead of a large and over-fitting hypersphere (the outer dashed circle with the radius b).

2.2: Scaling Condensation

To ensure that different localities are non-overlapping in the subspace, further condensation may be necessary while the condensation degree can be minimized. We condense a hypersphere from all directions equally such that the relative neighborhood relationship within a hypersphere is preserved and their condensation degrees are minimized. This process is called *scaling condensation*. Formally, this is a minimization of:

$$\sum_{i=1}^{m} \frac{R_i - r_i}{R_i},$$

with respect to r_i subject to

$$r_i + r_j \leq \mathrm{dist}(o_i, o_j),\ 0 \leq r_i \leq R_i,\ 1 \leq i, j \leq m,$$

where dist() is the Euclidean distance function, m is the number of hyperspheres, r_i is the radius of the R_i-hypersphere^D after further scaling condensation, and o_i is the projection of O_i in the d-dimensional subspace, which is computed by:

$$o_i = O_i \cdot \Phi_d,$$

where Φ_d represents the matrix containing 1^{st} to d^{th} principal components of the original data set. This is a convex optimization problem, which will be proved in Section 4.2.

After r_i is obtained, R_i-hypersphere^D is then scaled to a smaller r_i-hypersphere^D. That is, given a point X_i' in R_i-hypersphere^D, it is transformed as follows:

$$X_i = (X_i' - O_i)\frac{r_i}{R_i} + O_i.$$

It can be observed from the above transformation that the scaling condensation does not change the relative neighborhood relationship within the locality. Mapping the cluster centers onto the subspace based on the principal components of the original data set also preserves the global structure across localities.

3. **Map to Subspace**

Finally, for each D-dimensional point X_i after locality condensation, its d-dimensional projection x_i is:

$$x_i = X_i \cdot \Phi_d.$$

By changing d to different values, we can generate projections with different dimensionalities.

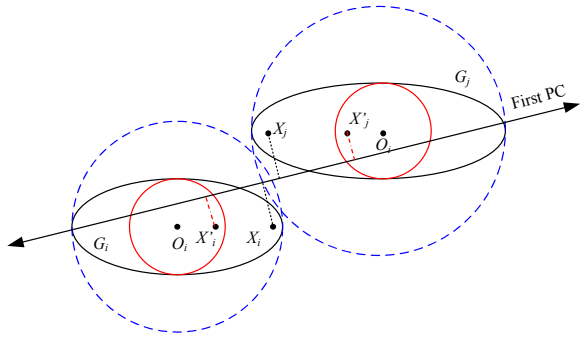


Figure 5: Effect of condensation

3.2 Illustrations

In Step 1, localities can be generated by some clustering methods with objective functions. The goal is to minimize the neighboring information loss among data points in different localities.

In Step 2, the core of our algorithm, each locality is condensed into a smaller hypersphere such that none of their projections in the subspace will overlap, with the optimization on condensation degrees.

In Step 2.1, elliptical condensation condenses a locality with an intrinsic elliptical distribution into a hypersphere (justified in Section 4.1). To ensure that the obtained hypersphere is smaller than the one with a radius of the hyperellipsoid along the largest axis, we further scale it to have a radius R_i equal to the radius of the hyperellipsoid along the second largest axis.

By elliptical condensation, the neighborhood relationship of data points within a locality is retained by the Mahalanobis distance. It implies that similarity search quality based on the Euclidean distance may be degraded. However, as a gain, the margins of different localities are enlarged since the size of each locality becomes smaller. As illustrated in Figure 5, localities G_i and G_j, signified with two ellipses, can be represented as two hyperspheres by preserving the original Euclidean distance (the outer dashed circles) or the condensed Mahalanobis distance (the inner circles). X_i and X_j are two distant points belonging to G_i and G_j respectively. Points X'_i and X'_j indicate the locations of X_i and X_j after applying elliptical condensation. After projecting them onto the 1-dimensional subspace, X'_i and X'_j remain distant to each other, while X_i and X_j tend to be closer. As a consequence, elliptical condensation could reduce the potential overlaps among different localities in the subspace. In other words, a higher discriminative capacity across different localities can be achieved.

Although elliptical condensation condenses each locality into a smaller hypersphere, it does not guarantee that their projections in the subspace do not overlap with each other. To avoid the invasion of different localities in the subspace, scaling condensation in Step 2.2 is further needed when necessary.

Finally, in Step 3, for each point after locality condensation, it is projected onto the chosen d-dimensional subspace by the top d principal components.

Obviously, the projection of a hypersphere by PCA is also a hypersphere with the same radius. Based on this, the projection of a hypersphere can be determined, together with its center's projection in the subspace, i.e., o_i. Note that mapping the cluster centers onto the subspace based on the principal components of the whole data set also preserves the global structure across localities. Therefore, based on the information in the subspace (i.e., o_i and R_i), our problem can be formalized as a convex problem (proved in Section 4.2) to avoid any potential overlap with the constraint that the condensation degree is minimized. By the low complexity interior-point methods that achieve optimization by going through the middle of the solid defined by the problem rather than around its surface [4], r_i can be obtained.

To have a clear picture, Figure 6 shows a running example of LC algorithm with the data set shown in Figure 1. Assume three localities of elliptical distributions are generated as shown in Figure 6(a). After elliptical condensation, each locality (hyperellipsoid) is condensed to a smaller hypersphere, as shown in Figure 6(b). Figure 6(c) shows that the projections of the • locality and the + locality on the first principal component of the data set have some overlap. Further scaling condensation is then performed on the larger • locality by convex optimization to avoid the invasion of different localities. As shown in Figure 6(c), the projections of three localities in the 1-dimensional subspace have no overlap. The global relationship across different localities is also preserved by their centers' projections in the subspace. Given a query from any locality, distant points from different localities are less likely to be included in the result set. Compared with PCA and LPP as shown in Figures 2 and 3, LC can avoid the overlap among different localities in the subspace.

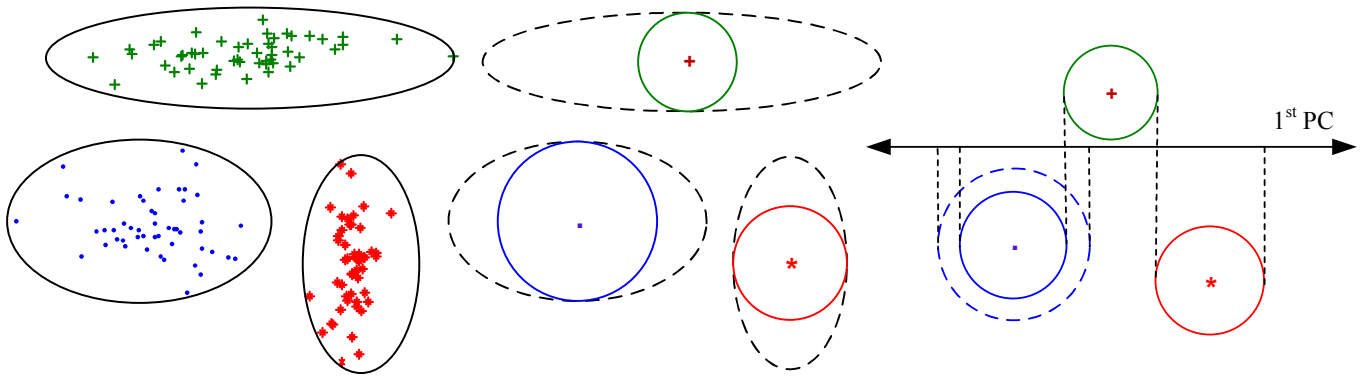
For top-k query processing, the projection of a given query Q in the subspace, denoted as q, is first derived by $q = Q \cdot \Phi_d$. A top-k retrieval for q in the d-dimensional subspace is then performed. Indexing structures [12, 19] can be built on the low-dimensional subspace to speed up the search.

In summary, LC achieves greater power in distinguishing points from different localities while the sacrifice on the relationship of boundary points can be minimized based on convex optimization.

3.3 Further Discussions

Parameter m, the number of localities generated, may affect the performance of LC. When m is too small, each locality is very large. During elliptical condensation, it is more likely that a point's neighbors within the locality measured by the Euclidean distance will be affected. On the other hand, when m is too large, there will be too many small localities. The probability of a point's neighbors to be separated into other localities becomes larger. The granularity of locality generation needs be studied to understand its effect on the performance, as we will see later in the experiments. Note that, at this stage, the k-means algorithm is used to determine the localities, but we do not exclude the use of other techniques according to the different data sets and data distributions.

Another important research issue is the selection of R_i in elliptical condensation. A too small value of R_i may significantly enlarge the margins of different localities, but also greatly affect the intra-locality neighborhood relationship in the Euclidean space. In this paper, we set R_i to be the radius of the hyperellipsoid containing G_i along its second largest axis since the intra-locality neighborhood relationship can be less affected by elliptical condensation while the



(a) Three elliptical localities (b) After elliptical condensation (c) After scaling condensation

Figure 6: Geometric illustration of LC

radius of each hypersphere is reduced. Other techniques can also be designed for determining R_i.

The condensation of localities may also have certain side effects. Elliptical condensation maintains the neighborhood relationship *within* the locality based on the Mahalanobis distance, which could be different from the original relationship based on Euclidean distance, and enlarging the margin of different localities may also affect the results of queries lying on locality boundaries. However, the achievement of eliminating the overlaps among different localities in the subspace can prevent most, if not all, distant points which are totally irrelevant to the query from being included in the result set. As demonstrated in the experiments, the gain of our LC proposal is overwhelming, especially when the dimensionality of subspace is quite low.

4. JUSTIFICATIONS

In this section, we provide the justifications to prove the correctness of our algorithm.

4.1 Elliptical Condensation

The Mahalanobis distance maps all points on an ellipsoid to the same distance to the origin by utilizing the covariance matrix. The covariance matrix of a locality measures the relative range of the locality along each dimension by variance values, and indicates the direction of orientation by covariance values. The Mahalanobis distance differs from the Euclidean distance in that it takes the correlations of a data set into account by giving less weight to directions with high variability than directions with low variability when determining the distance from a data point to the center of the data set. Given a data point X_i and the center O_i of a locality G_i, the Mahalanobis distance from X_i to O_i is defined as:

$$\text{dist}_{\text{Maha}}(X_i, O_i) = \sqrt{(X_i - O_i)M_i(X_i - O_i)^T},$$

where M_i is the inverse of the covariance matrix of G_i. Only if $M_i = I$, the Mahalanobis distance is the same as the Euclidean distance. According to the Singular Value Decomposition theorem, M_i can be decomposed into:

$$M_i = U_i S_i V_i^T,$$

where the columns of U_i are made up of the eigenvectors of $M_i M_i^T$, the columns of V_i are made up of the eigenvectors of $M_i^T M_i$, and S_i is a diagonal matrix containing square roots of eigenvalues from $M_i M_i^T$. Since M_i is symmetric, it is obvious that $M_i M_i^T$ is equal to $M_i^T M_i$, U_i is equal to V_i and S_i is a diagonal matrix. Therefore, we can get a further decomposition of M_i is terms of:

$$\begin{aligned} M_i &= U_i S_i V_i^T \\ &= U_i \sqrt{S_i} \sqrt{S_i} U_i^T \\ &= (U_i \sqrt{S_i})(U_i \sqrt{S_i})^T. \end{aligned}$$

Let $A_i = U_i \sqrt{S_i}$, then the original Mahalanobis distance can be rewritten as:

$$\begin{aligned} \text{dist}_{\text{Maha}}(X_i, O_i) &= \sqrt{(X_i - O_i)A_i A_i^T (X_i - O_i)^T} \\ &= \sqrt{((X_i - O_i)A_i)((X_i - O_i)A_i)^T}. \end{aligned}$$

Thus, A_i can be considered as a linear transformation on the original points. Given a point X_i, mapping X_i to X_i^{m} as $(X_i - O_i)A_i$ transforms the distance space from Mahalanobis to Euclidean, i.e., from elliptical to spherical.

Note that so far the points have been mapped into a new coordinate system whose axes are the eigenvectors in U_i. We have to transform the points back into the original coordinate system simply by:

$$X_i' = X_i^{\text{m}} U^{-1} + O_i.$$

By the above transformation, each locality with the intrinsic elliptical distribution can be transformed into a hypersphere in the original coordinate system.

4.2 Convex Optimization

The convex optimization problem has been well studied in mathematics. It has a wide range of applications in combinatorial optimization and global optimization, where it is used to find bounds on the optimal value, as well as approximate solutions. Typical applications include automatic control systems, signal processing, networking, electronic circuit design, data analysis and modelling, and finance. There are great advantages to recognize or formulate a problem as a convex optimization problem. The most basic advantage is that the problem can then be solved, very reliably and efficiently, using interior-point methods or other special methods for convex optimization [4].

A convex optimization problem is of the form to minimize $f_0(r)$ subject to $f_i(r) \leq b_i$, $i = 1, \ldots, c$, where c is the number of constraint functions. The functions $f_0, \ldots, f_c$:

$\mathbb{R}^m \rightarrow \mathbb{R}$ are convex, i.e., they satisfy

$$f_i(\alpha r + \beta r') \leq \alpha f_i(r) + \beta f_i(r'),$$

for all r, $r' \in \mathbb{R}^m$ and all α, $\beta \in [0, 1]$ with $\alpha + \beta = 1$. In the convex optimization problem, f_0 is the objective function, r is the optimization variables, $f_1, \ldots, f_c$ are constraint functions. The optimal solution $r = (r_1, \ldots, r_m)$ has the smallest value of f_0 among all vectors that satisfy the constraints. In our problem, the objective function f_0 and constraint functions can be formally defined as:
minimize

$$f_0(r_i) = \sum_{i=1}^{m} \frac{R_i - r_i}{R_i},$$

subject to

$$r_i + r_j \leq \text{dist}(o_i, o_j),\ r_i \leq R_i,\ -r_i \leq 0,\ 1 \leq i, j \leq m,$$

where m is the number of hyperspheres.

As an affine function of r_i, $f_0(r_i)$ is convex. Similarly, each constraint function is also convex. Therefore, the problem can be solved as a convex optimization problem using interior-point methods [4].

5. EXPERIMENTS

In this section, we report the results of an extensive performance study conducted to evaluate our LC dimensionality reduction method on large real-life image data sets.

5.1 Set Up

Three real-life image data sets are used in our experiments. The original dimensionalities of their feature spaces range from 32 to 159.

- **Corel image set.** This image set is widely used and contains 68,040 Corel images of various categories[1]. The feature of 32-dimensional HSV color histogram is used in our experiments.
- **Getty image set.** This image set contains 21,820 images downloaded from Getty image archive website[2]. The feature of 128-dimensional RGB color histogram is used in our experiments. The list of Getty image IDs used to make up the data set [29] can be downloaded[3].
- **WWW image set.** We also created one WWW image set consisting of 48,581 images randomly crawled from over 40,000 Web pages. In our experiments, the feature of 159-dimensional HSV color histogram[4] is used.

For dimensionality reduction in similarity search applications, it is ideal that top-k retrieval (or k-nearest neighbor search) in the low-dimensional subspace can yield the same results as those in the original high-dimensional space [8, 24]. Similarly, we evaluate the quality of dimensionality reduction by *precision*, which is defined as:

$$\text{precision} = \frac{|\text{result}^d \cap \text{result}^D|}{k},$$

where result^d and result^D are the results returned from the d-dimensional subspace and D-dimensional original space respectively, k is the number of results returned, and $|\cdot|$ represents the size. Only top-k results are considered in our evaluation. Note that semantic judgement is not considered for our content-based image similarity search here. Standard Euclidean distance is used to measure the dissimilarity between two image feature vectors for top-k ranking.

In this paper, we mainly focus on the quality evaluation of dimensionality reduction, since an indexing structure can be easily deployed to index the reduced subspace and the efficiency gain achieved by dimensionality reduction has been demonstrated in many existing work. Since LC is a global dimensionality reduction approach, we compare LC with the classical global methods PCA and LPP which are applicable for top-k retrieval. In LPP, 100 nearest neighbors for each point are used in constructing its adjacent graph. The results are reported by averaging 100 queries randomly selected from the data sets.

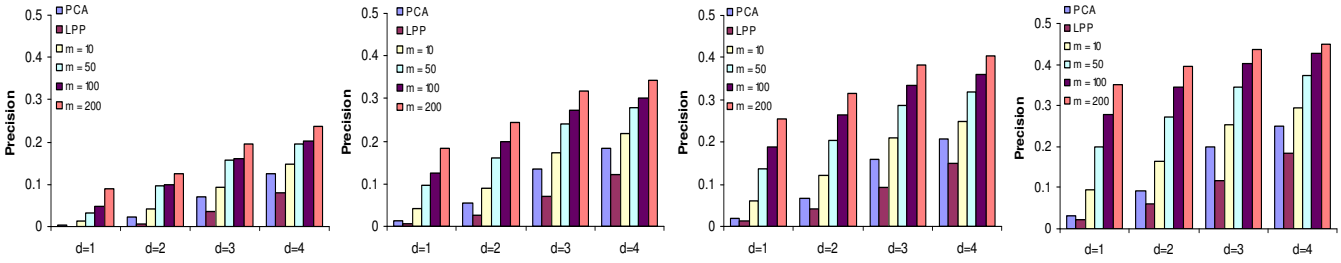
5.2 Results Analysis

In our experiments, we test the effect of different m values in our LC method and the effect of different k values in top-k image retrieval, compared with PCA and LPP. The values of d (i.e., the dimensionality of a reduced subspace) range from 1 to 4 so that the subspace can be effectively indexed by existing multi-dimensional indexing structures for facilitating similarity search [12, 19].

Figure 7 shows the results of three methods on Corel image set, where four different values for the number of localities in LC (i.e., m=10, 50, 100 and 200) and four different k values (i.e., k=10, 50, 100 and 200) are tested. From Figure 7(a) when k=10, three observations can be made. First and most importantly, for different d values, our LC method outperforms PCA significantly on all tested m values, and PCA in turn outperforms LPP. The relative improvement gaps achieved by LC over PCA and LPP are much greater in the lower dimensional subspace where the overlaps among different localities are more noticeable. It proves that the gain of LC to prevent different localities from being overlapped in the subspace is much greater than the loss of intra-locality and boundary neighborhood relationship caused by condensation. This confirms the superiority of LC over PCA and LPP in distance-based top-k image retrieval. The reason of the bad performance of LPP, we believe, is that LPP also encourages distant points to be mapped nearby in the subspace in order to minimize its objective function as mentioned earlier. Second, as the number of localities in LC (i.e., m) increases, the improvements achieved by LC over PCA and LPP become even larger. For example, when the original 32-dimensional Corel image feature space is reduced to a 4-dimensional (i.e., d=4) subspace, LC with m=10, 50, 100 and 200 achieves a precision of 15%, 19%, 20% and 24% respectively, while PCA and LPP have precisions of around 12% and 8% respectively. Note that a larger m value leads to smaller localities. Since k is much smaller than the average locality size (which can be computed by dividing the total data set size over m), top-k retrieval within a smaller locality is expected to be more accurate. Third, as the dimensionality of subspace (i.e, d) increases, the precision for each method also goes up. This is clear since more information can be preserved for a higher dimensional subspace.

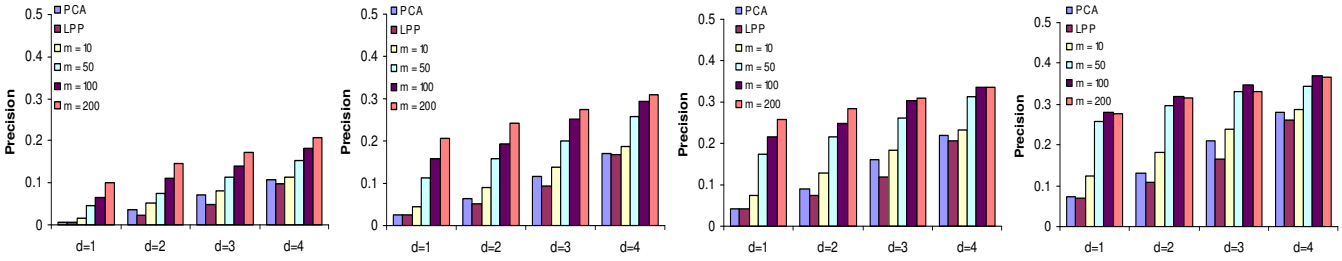
Figure 7(b), Figure 7(c) and Figure 7(d) report the results

[1] http://kdd.ics.uci.edu/databases/CorelFeatures
[2] http://creative.gettyimages.com
[3] http://mmir.doc.ic.ac.uk/www-pub/civr2005
[4] http://www.itee.uq.edu.au/~shenht/histogram159.rar



(a) $k = 10$ (b) $k = 50$ (c) $k = 100$ (d) $k = 200$

Figure 7: Results on the Corel image set



(a) $k = 10$ (b) $k = 50$ (c) $k = 100$ (d) $k = 200$

Figure 8: Results on the Getty image set

for larger k=50, 100 and 200 respectively on the Corel image set. Clearly, for different k values, similar patterns to that of Figure 7(a) can also be observed. At the same time, as k increases, the precision for each method also rises. This is reasonable since the precision computation is less sensitive to a larger k.

Figure 8 compares the results of three methods on Getty image set, with the same setting as on Corel image set. Similarly, our LC method improves PCA greatly on all tested m values and PCA in turn outperforms LPP. Especially for small d values (i.e., d=1 or 2), the precisions of PCA and LPP are mostly less than 10%, which are too low to be acceptable for retrieval. Meanwhile, the precision for each method increases as d or k increases. However, the effectiveness of LC does not always improve as the number of localities (i.e., m) grows. As can be seen from Figure 8(d) when k=200, LC performs best when m=100. Note that the average locality size for Getty image set is $21,820/m$. When k=200, a large m value such as 150 or 200 leads to a smaller average locality size than k. As a result, more localities are expected to be searched for top-k results. Since LC enlarges the margins among different localities, its effectiveness potentially deteriorates when top-k results are located in more localities. Therefore, a proper m value should be set for LC to achieve better performance. From Figure 8, it is suggested that the average size of localities in LC should be around the value of k, i.e., we recommend that $m \cdot k$ should be about the data set size.

Figure 9 shows the results of three methods on WWW image set, with the same setting. It further confirms the effectiveness of LC. From Figure 7, Figure 8 and Figure 9, we can affirm the superiority consistence of LC with different m values over PCA and LPP for different k values in different d-dimensional subspaces and different data sets.

In summary, for the coin of dimensionality reduction, since PCA aims at preserving the global variance and LC aims at preserving the local neighborhood structure in the subspace, LC considers both sides of the coin and provides a more reliable dimensionality reduction for top-k image retrieval.

6. CONCLUSIONS

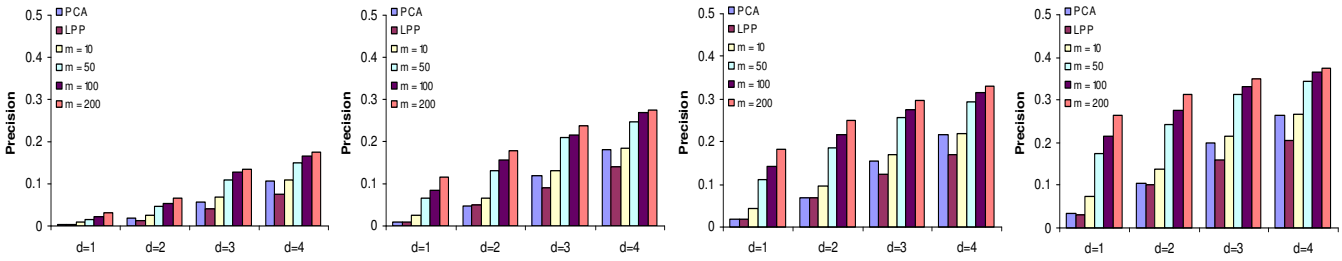
This paper presents a new dimensionality reduction method called Locality Condensation (LC) to simplify complex data by finding a smaller number of dimensions that can represent a large number of original dimensions. LC preserves the similarity information reliably at the same time preventing distant points from invading other neighborhoods. By elliptical condensation and scaling condensation, nearby points remain nearby while distant points still remain distant, suggesting that the similarity search can be performed more accurately in the subspace. Empirical studies are conduced on three image data sets and the results validate that the proposed method is superior to two existing proposals for dimensionality reduction. In future, the idea of LC can be further studied in other application domains, such as classification and pattern recognition.

Acknowledgments: This work is supported by an Australian Research Council (ARC) grant DP0663272 and European Union Sixth Framework Programme (FP6) through the integrated project Pharos (IST-2006-045035).

7. REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions



(a) $k = 10$ (b) $k = 50$ (c) $k = 100$ (d) $k = 200$

Figure 9: Results on the WWW image set

for lle, isomap, mds, eigenmaps, and spectral clustering. In *NIPS*, 2003.

[3] C. Böhm, S. Berchtold, and D. A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv.*, 33(3):322–373, 2001.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[5] D. Cai and X. He. Orthogonal locality preserving indexing. In *SIGIR*, pages 3–10, 2005.

[6] D. Cai, X. He, and J. Han. Spectral regression: a unified subspace learning framework for content-based image retrieval. In *ACM Multimedia*, pages 403–412, 2007.

[7] K. Chakrabarti, E. J. Keogh, S. Mehrotra, and M. J. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Trans. Database Syst.*, 27(2):188–228, 2002.

[8] K. Chakrabarti and S. Mehrotra. Local dimensionality reduction: A new approach to indexing high dimensional spaces. In *VLDB*, pages 89–100, 2000.

[9] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *NIPS*, pages 705–712, 2002.

[10] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[11] C. Elkan. Using the triangle inequality to accelerate k-means. In *ICML*, pages 147–153, 2003.

[12] V. Gaede and O. Günther. Multidimensional access methods. *ACM Comput. Surv.*, 30(2):170–231, 1998.

[13] X. He, D. Cai, and J. Han. Learning a maximum margin subspace for image retrieval. *IEEE Trans. Knowl. Data Eng.*, 20(2):189–201, 2008.

[14] X. He, D. Cai, H. Liu, and W.-Y. Ma. Locality preserving indexing for document representation. In *SIGIR*, pages 96–103, 2004.

[15] X. He, W. Min, D. Cai, and K. Zhou. Laplacian optimal design for image retrieval. In *SIGIR*, pages 119–126, 2007.

[16] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.

[17] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[18] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley and Sons, 2001.

[19] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang. idistance: An adaptive b^+-tree based indexing method for nearest neighbor search. *ACM Trans. Database Syst.*, 30(2):364–397, 2005.

[20] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, New-York, second edition, 2002.

[21] D. A. Keim and B. Bustos. Similarity search in multimedia databases. In *ICDE*, page 873, 2004.

[22] K. Mikolajczyk and J. Matas. Improving descriptors for fast tree matching by optimal linear projection. In *ICCV*, 2007.

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

[24] H. T. Shen, X. Zhou, and A. Zhou. An adaptive and dynamic dimensionality reduction method for high-dimensional indexing. *VLDB J.*, 16(2):219–234, 2007.

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

[26] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of The Royal Statistical Society Series B*, 61(3):611–622, 1999.

[27] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, pages 194–205, 1998.

[28] Y.-L. Wu, D. Agrawal, and A. E. Abbadi. A comparison of dft and dwt based similarity search in time-series databases. In *CIKM*, pages 488–495, 2000.

[29] A. Yavlinsky, E. Schofield, and S. M. Rüger. Automated image annotation using global features and robust nonparametric density estimation. In *CIVR*, pages 507–517, 2005.

[30] F. W. Young and R. M. Hamer. *Multidimensional Scaling: History, Theory and Applications.* Erlbaum, New York, 1987.

[31] J. Yu and Q. Tian. Learning image manifolds by semantic subspace projection. In *ACM Multimedia*, pages 297–306, 2006.

[32] X. Zheng, D. Cai, X. He, W.-Y. Ma, and X. Lin. Locality preserving clustering for image database. In *ACM Multimedia*, pages 885–891, 2004.