Dimensionality Reduction

Manoranjan Dash

Department of Information Systems

School of Computer Engineering

Nanyang Technological University

Nanyang Avenue, Singapore 639798

asmdash@ntu.edu.sg

Huan Liu

Department of Computer Science and Engineering

Arizona State University

PO BOX 878809, Tempe, AZ 85287-8809

hliu@asu.edu

Abstract

Dimensionality reduction studies methods that effectively reduce data dimensionality for efficient data processing tasks such as pattern recognition, machine learning, text retrieval, and data mining. We introduce the field of dimensionality reduction by dividing it into two parts: feature extraction and feature selection. Feature extraction creates new features resulting from the combination of the original features; and feature selection produces a *subset of the original features. Both attempt to reduce the dimensionality of a dataset in order to facilitate efficient data processing tasks. We introduce key concepts of feature extraction and feature selection, describe some basic methods, and illustrate their applications with some practical cases. Extensive research into dimensionality reduction is being carried out for the past many decades. Even today its demand is further increasing due to important high-dimensional applications such as gene expression data, text categorization, and document indexing.*

Keywords: dimensionality, feature selection, feature extraction.

1. Introduction

A dimension refers to a measurement of a certain aspect of an object. Dimensionality reduction is the study of methods for reducing the number of dimensions describing the object. Its general objectives are to remove irrelevant and redundant data to reduce the computational cost and avoid data over-fitting [52], and to improve the quality of data for efficient data-intensive processing tasks such as pattern recognition and data mining. Dimensionality reduction is an effective solution to the problem of "curse of dimensionality". When the number of dimensions increases linearly, experiments have shown that the required number of examples for learning increases exponentially [3]. Figure 1 shows an example of the curse of dimensionality.

In practice, researchers and practitioners interchangeably use dimension, feature, variable, and attribute. Similarly, we will interchangeably use object, example, vector, and instance. Consider an application in which a system processes data (speech signal, images, or patterns in general) in the form of a collection of vectors. For a particular application, it is more often than not that a subset of features is relevant and in some cases, a large number of features are irrelevant. This problem can be caused by factors such as: (1) many dimensions will have variation smaller than the measurement noise and thus will be *irrelevant*, and (2) many dimensions will be correlated (through linear combinations or functional dependence) to others and thus will be *redundant*. Therefore, in many situations, it is recommended to remove the irrelevant and redundant dimensions, producing a more economical

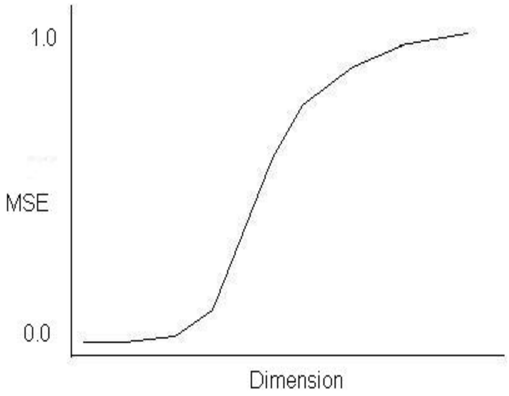

Figure 1. An example of curse of dimensionality: MSE is the mean squared error of an 1-nearest neighbor rule [19, 33]. Each dimension is generated uniformly on [-1, 1]. As the dimensionality increases the MSE increases very sharply until it levels off at 1.0. This happens as early as dimensionality = 10. See [20] for further details.

representation of the data [16].

Dimensionality reduction is a research area at the intersection of several disciplines, including statistics, databases, data mining, text mining, pattern recognition, machine learning, artificial intelligence, visualization and optimization. Each of these areas has its own way of looking at the problem. For example, in pattern recognition the problem of dimensionality reduction is to extract a small set of features that recovers most of the variability of the data. In text mining, however, the problem is defined as selecting a small subset of words or terms (not new features that are combination of words or terms). Use of this important technique also varies with the application domain. Examples of applications of dimensionality reduction techniques include: mining of text documents, gene structure discovery, image processing, statistical learning, and exploratory data analysis. Different applications need to be treated with different techniques. Depending on the application, new features may be extracted as in the case of exploratory analysis, or a small subset of original features are selected as in the case of gene structure discovery.

Dimensionality reduction has been a subject of much research currently and over the past several decades (some good overviews are available [44, 24, 13, 22]). Especially the pioneering work of Sammon [18] is giving inspiration for today's information processing systems. Sammon, in the late 70's, combined dimensionality reduction with the issues such as classification, and interactive visual data analysis. Recently, there is a renewed interest in this topic due to massive data of large dimensionality created in data mining, data warehousing, and knowledge discovery applications. Other applications such as genome project, text mining, and web mining, also require efficient dimensionality reduction methods.

Dimensionality reduction methods can be grouped in various ways: (1) feature selection or feature extraction, (2) linear or nonlinear, (3) supervised or unsupervised, and (4) local or global. Dimensionality reduction methods are often classified into feature selection or feature extraction. In feature selection, a subset of original features are selected in the end. In feature extraction, new features are extracted using some mapping (linear or nonlinear) from the original set of features. Linear methods such as principal components analysis (PCA) use a linear mapping to extract new features from original features [13]. Similarly, nonlinear methods such as Sammon's mapping [44], locally linear embedding [42], and ISOMAP [49] use a non-linear mapping to extract new features. Supervised methods can take advantage of any class information present in the data whereas unsupervised methods do not use this class information. One limitation of the supervised methods is that characteristic variables that describe examples of infrequent classes tend to be easily removed as a result of dimensionality reduction making use of the class distribution. Typically, supervised dimensionality reduction methods can be further divided into local or global methods. In a local method, features are selected for each category of the class feature; in case of a global method, features are selected for all categories. Among all these different ways of categorizing dimensionality reduction methods, we will mainly describe various methods of dimensionality reduction methods in terms of feature extraction or feature selection.

In the following two sections, we introduce the basic concepts and key techniques of feature extraction and selection, respectively. We then discuss some dimensionality reduction methods in practice in Section 4.

2. Feature Extraction

Feature extraction can be defined as follows: Given a set of features $S = \{v_1, v_2, ..., v_D\}$, find a new set of features S' derived from a linear or non-linear mapping of S. The cardinality of $|S'| = d$ and $J(S') \geq J(T)$ for all derived set of features T with $|T| = d$, where J is the evaluation function. Here d or some other parameter that can determine d (e.g., a threshold eigen value) is usually specified by the user.

When all existing features are recombined to yield new features then we are dealing with feature extraction. Hence a mapping is defined that transforms any original D dimensional feature vector to a new d dimensional feature vector. Ideally the mapping conserves or even enhances the discriminatory information while simultaneously reducing the dimensionality of the feature vector. Mapping can be linear or nonlinear. Figure 2 depicts this pictorially. The following descriptions of a sample of classical feature extraction methods will bring out the methodical difference between feature transformation from selection.

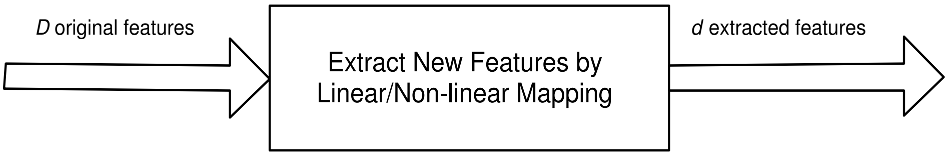

Figure 2. Feature extraction process

Principal components analysis (PCA) It is the most widely used linear feature extraction method [23]. It is also called Karhunen-Loeve transform in signal processing literature. The class information is not taken into consideration in this method. The objective of this method is to find a set of d orthogonal basis vectors that maximally captures the relationship between the original dimensions. It can be shown that the j^{th} principal component direction is along an eigen vector direction of the global covariance matrix $C = \frac{1}{N-1}\sum_{j=1}^{N}[(x_j - \mu)^T(x_j - \mu)]$ of the feature vector x and its global mean μ. The first d eigen vectors, e_j, then define a D X d matrix M with the eigen vectors as columns that transforms the original sample x to the extracted sample $y = M^T x^T$.

The eigen analysis can be done using standard mathematical softwares, e.g. [39]. PCA is an information conserving transform. Using all d' $(d \leq d' \leq D)$ non-zero eigen vectors conserves the information contained in the orthogonal features. The new features are orthogonal to each other and there is no covariance (and correlation) between any two features. The variance of the new feature y_j is the eigen value λ_j. The original feature vector can be reconstructed as $x = My^T$. The truncation of those eigen vectors, which are associated with the smallest eigen values, does not incur a large information loss (approximate loss: $\epsilon = \sum_{j=d+1}^{d'}\lambda_j$). PCA can therefore be used as an information conserving, correlation eliminating and dimensionality reduction feature extraction method.

The process of determining most influential features having maximum eigen values is called singular value decomposition (SVD). Another method similar to PCA is called latent semantic indexing (LSI). LSI has been successfully used in information retrieval for clustering documents. In Section 4 we give brief descriptions of some applications using PCA and LSI.

Linear discriminative analysis (LDA) Unlike PCA, LDA considers the class information. The only difference between PCA and LDA is the matrix that is considered. LDA uses a within-scatter matrix of all c classes: $S_W = \sum_{i=1}^{c}\sum_{j=1}^{N_i}[(x_j-\mu_i)^T(x_j-\mu_i)]$, and a between-scatter matrix: $S_B = \sum_{i=1}^{c}[(\mu_i-\mu)^T(\mu_i-\mu)]$, where N_i is the number of objects within class i, μ_i is the common mean of class i, and μ is the mixture mean of all classes. Then at most $d = c-1$ non-zero eigen vectors, associated to the largest eigen values of the matrix $S_W^{-1}S_B$, define the D X d feature extraction matrix M. This transformation maximizes the between-class scatter while minimizing the within-class scatter (i.e., $maximize\frac{det(S_B)}{det(S_W)}$) where $det(.)$ denotes determinant of a square matrix. Such a transformation should retain class separability while reducing the variation due to other sources. The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible. It seeks to find directions along which the classes are best separated. LDA works well if the data has a multivariate normal distribution.

Sammon map is an example of a non-linear feature extraction method, whereas the above methods are all linear [44]. It is mainly used for 2-D visualization of high-dimensional data. The non-linear mapping can be performed for any dimensionality $d < D$. Sammon's algorithm uses gradient descent technique to minimize an error function in order to map from D to d dimensions. The error function is given as

$$E = \frac{1}{\sum_{j=1}^{N-1}\sum_{k=j+1}^{N}\delta_{jk}}\sum_{j=1}^{N-1}\sum_{k=j+1}^{N}\frac{\delta_{jk}-{d_{jk}}^2}{\delta_{jk}}$$

where δ_{jk} is the distance between two points in the extracted d-dimensional space, and d_{jk} is the distance between two points in the original D-dimensional space. The mapping attempts to fit N points in the lower-space, such that their inter-point distances approximate the corresponding distances in the higher-space. In [9] linear feature extraction method PCA is compared with non-linear methods such as Sammon's mapping, multi-dimensional scaling [27] and self-organizing mapping [25] using texture data. Results clearly show that classification performance improves, particularly for small values of d, by using non-linear methods compared to that of linear methods such as PCA. The reason being, performing non-linear feature extraction provides a better characterization of the data

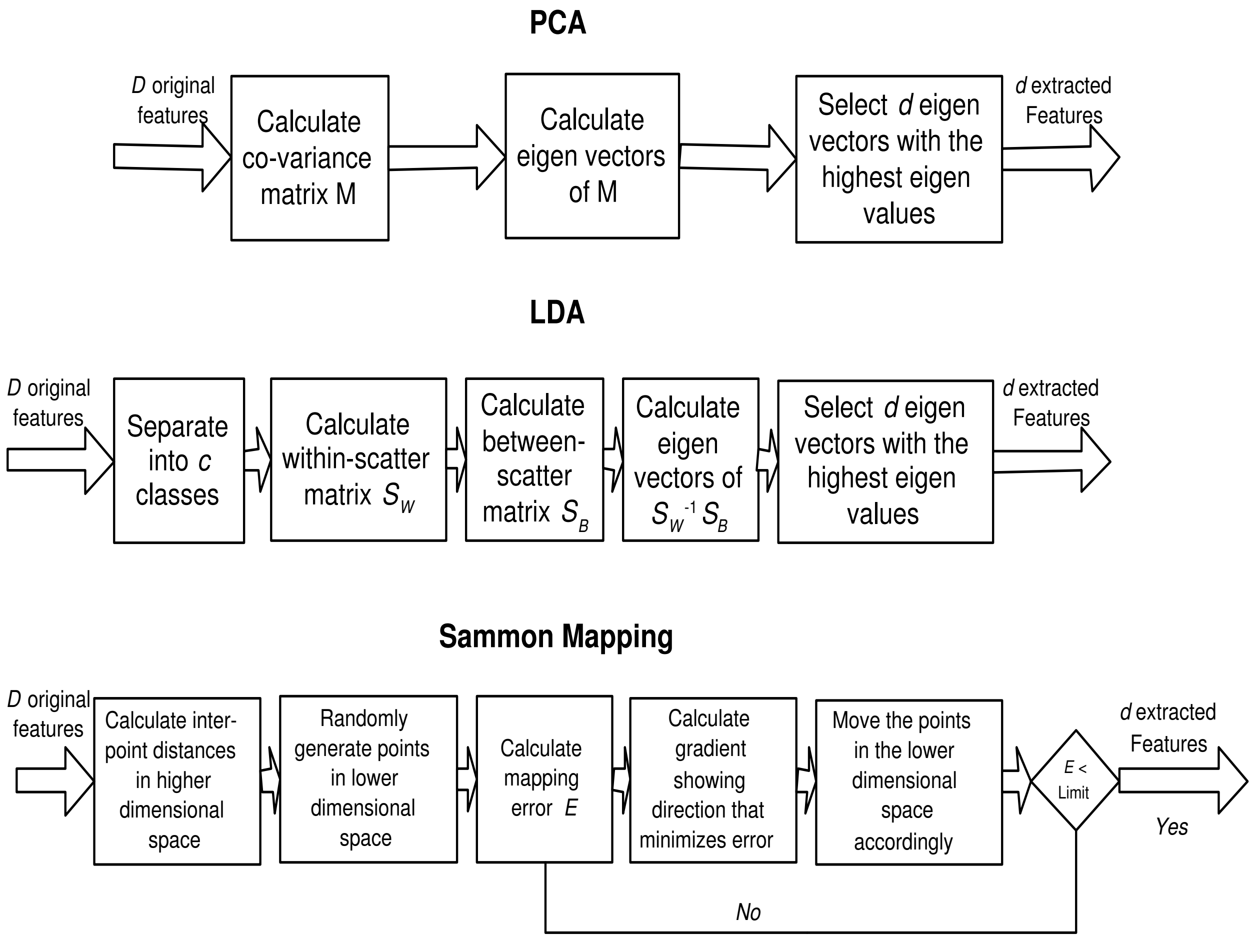
PCA
D original features
Calculate co-variance matrix M
Calculate eigen vectors of M
Select d eigen vectors with the highest eigen values
d extracted Features
LDA
D original features
Separate into c classes
Calculate within-scatter matrix S_W
Calculate between-scatter matrix S_B
Calculate eigen vectors of $S_W^{-1} S_B$
Select d eigen vectors with the highest eigen values
d extracted Features
Sammon Mapping
D original features
Calculate inter-point distances in higher dimensional space
Randomly generate points in lower dimensional space
Calculate mapping error E
Calculate gradient showing direction that minimizes error
Move the points in the lower dimensional space accordingly
E < Limit
d extracted Features
Yes
No

Figure 3. Procedures for PCA, LDA and Sammon Mapping

in a smaller number of features.

3. Feature Selection

Feature selection can be defined as follows: Given a set of features $S = \{v_1, v_2, ..., v_D\}$, find a subset S' of S with $|S'| = d$ such that $J(S') \geq J(T)$ for all $T \subset S$, $|T| = d$ where J is the evaluation function. Here d is usually specified by the user.

A feature selection algorithm requires the following ingredients: a generation or search strategy, an evaluation method, a stopping criterion and/or a validation method [7, 30, 31, 55]. See Figure 4 for a block diagram showing

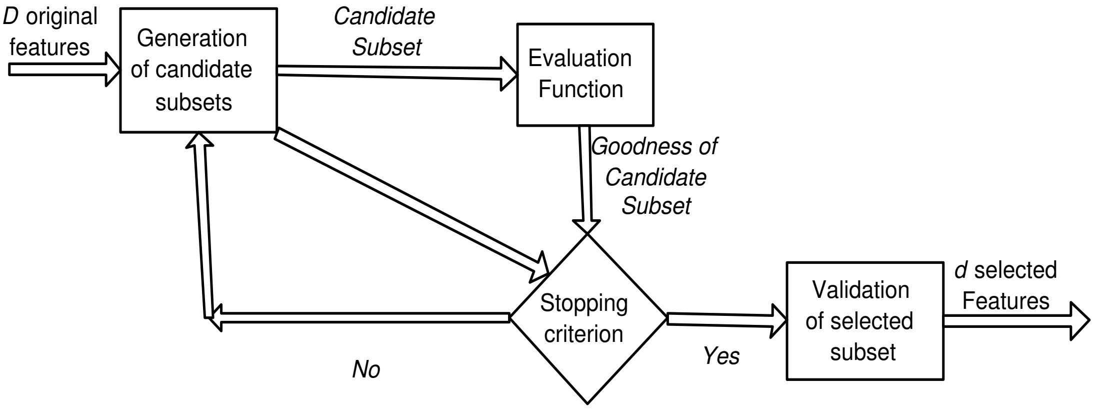

Figure 4. Feature selection process

relationships between these components. The search or generation strategy decides the way how combinations of features are tested for a certain goodness. Since exhaustive search is usually prohibitive, alternative strategies must be employed. The evaluation or selection function assesses the goodness of a set of features and provides a ranking possibility for the selection process. The stopping criterion is of less importance. Usually a predefined number of features to be selected decides the stopping of search procedures. Validation itself is not part of the selection process but it is nonetheless carried out to check the validity of the selected features. In the following we briefly discuss the most important search strategies and evaluation criteria.

Generation/Search Procedure With total D features there are 2^D candidate feature subsets to be searched. This is a huge number even for moderate D. In the literature there are different approaches for solving this problem, namely: complete, heuristic and random.

Complete Search: Schlimmer [45] argues that just because the search must be complete does not mean that it must be exhaustive. Different heuristic functions are used to reduce the search space without jeopardizing the chances of finding the optimal subset. Branch and bound method [35] is one such method that guarantees optimality if the features obey monotonocity. Unfortunately many often-used evaluation criteria such as classifier error rate are not monotonic.

Heuristic Search: In each iteration of this method all remaining features yet to be selected are considered for selection. This is called sequential forward selection. When the search is backward it is called sequential backward selection. There are also combinations of these two approaches such as beam search [11].

Random Search: It searches randomly and usually stops after a maximum number of iterations. This has advantages over heuristic method in that, unlike heuristic methods, it is less likely to be trapped in local optima [32].

Evaluation Functions An evaluation function or selection criterion $J()$ aims at finding the best set of features in the reduced dimensionality d from the set of all features. Hence the best set S' maximizes the criterion function over all other possible combinations of d features. Some important evaluation functions are briefly described here.

Distance Measure: It is also known as separability, divergence, or discrimination measure. For a two class problem, a feature f_i is preferred to another feature f_j if f_i induces a greater difference between the two-class conditional probabilities than f_j [26].

Information Measure: These measures typically determine the information gain from a feature which is the difference between the prior uncertainty and expected posterior uncertainty using the feature. The feature giving higher information gain is selected [40].

Dependence Measures: It quantifies the ability of a feature to predict the value of the class variable. An example is correlation coefficient. If feature f_i has a higher correlation with the class variable than feature f_j, then feature f_i is preferred [34].

Consistency Measures: These measures prefer a consistent hypothesis definable over as few features as possible. A feature set is consistent if for the same set of values for the feature set the class variable does not change [32].

Classifier Error Rate Measure: The above four types of criteria are typically known as *filter* type while classifier error rate is known as *wrapper* type. The classifier that will be used after feature selection is also used to select the features. The feature set giving the minimum classifier error rate is selected. More details on wrapper methods are given in [4].

Feature Selection for Unsupervised Learning The above discussion is mostly for supervised learning where class information is available. Lately, feature selection has been attempted for unsupervised learning, and amongst different unsupervised learning it has been applied mostly to clustering and visualization.

In the last several years a number of methods for feature selection for clustering are proposed most of which are wrapper in approach. Here a clustering algorithm is used to evaluate the candidate feature subsets. Wrapper methods can be categorized based on whether they select features for the whole data (global type) or for each cluster separately (local type). The global type assumes a subset of features to be more important than others for the whole data while the local type assumes each cluster to have a subset of important features. In case of global type a feature selection method is run over the whole data whereas for local type first clustering is done over the data using all features and then important features are selected for each cluster separately using a feature selection method.

Selecting a set of features for unsupervised learning such as clustering is arguably more difficult than selecting for supervised learning such as classification because of the absence of any class information in the former. This is also the reason for extensive research being conducted for feature selection for classification compared to that for clustering. The difficult part is to evaluate the candidate subsets and compare against each other in order to select the optimal subset of features. First of all quantifying the quality of clustering is far less straight forward and less accurate than classification. On top of it, one must compare the quality of clustering across varying dimensionality in order to select the optimal subset of features. So, one requires evaluation methods that are invariant to varying dimensionality. Examples of such methods used in various research work are: trace measure [8, 14], visualization [14, 15], ranking of features and user selects a number of most important features [8, 10, 14, 17, 47, 48], Bayesian statistical estimation framework [50], and entropy [8, 6]. These are global type. Examples of local methods are Manhattan distance [1], and dense regions [2].

4. Dimensionality Reduction in Practice

In this section we discuss some applications of feature extraction and selection methods.

4.1 Uses of Feature Extraction Methods

Uses of PCA in Regression Analysis PCA can be used in regression analysis in a number of ways [13]. If the independent variables are highly correlated, then they can be transformed to principal components (PCs) and the PCs can be used as the independent variables. If we do not want to transform the independent variables, then the PCs can be used indirectly to improve the precision of the regression parameter estimates associated with the independent variables. PCA can also be used as a diagnostic tool to detect multi-colinearities among the independent variables. Multi-colinearity means that one or more independent variables are essentially linear combinations of other independent variables.

Using PCs to detect Outlying and Influential Observations A major advantage of PCA is that if the first two PCs account for a substantial portion of the total variation, then we can approximate the distribution of the observations in the variable space by plotting the PCs [13]. This 2-dimensional representation of the D-dimensional observations can be used in a number of ways. The plot can be examined for outlying observations, for influential observations, or it can be used to see if the observations can be visually clustered. Outlying observations are observations that lie at a considerable distance from the bulk of the observations or do not conform to the general pattern the observations exhibit. Outlying observations are called influential observations if their deletion from a particular analysis leads to different results.

Use of PCs in Cluster Analysis If the first two or three PCs account for a substantial proportion of the total variation, then we can also use the plots to visually identify clusters [13]. A *cluster* is a group of observations that are “closer” to each other than they are to observations in other clusters or groups. There are a large number of clustering algorithms that are used to cluster data [21]. There is no significant advantage in transforming the original observations to principal components prior to the clustering since the same information is contained in the original and the transformed data. That is, for any distance function, the distances among examples computed from principal components are equal to the corresponding distances computed from the original variables using an equivalent but different distance function. The only advantage of employing PCs in cluster analysis is to be able to plot the components and visually search for clusters of observations. PCs can also be used to verify the clusters determined on the basis of another clustering algorithm. We can see if the defined clusters are homogeneous, distinct, and aesthetically appealing to the eye. Clustering algorithms will define clusters even if none exist, i.e. even if the observations are evenly spread throughout the variable space. For this reason, a plot of the data on the first two PCs can be informative if they account for a large portion of the total variance.

Application to Computer Vision PCA is used in computer vision to find patterns and to compress the images [12].

PCA for Finding Patterns: An example of its application to face recognition is as follows. Say we have 20 images. Each image is H pixels high by W pixels wide. For each image we can create an image vector of HXW dimensions. We can then put all the images together in one big image-matrix like this:

$$\begin{pmatrix} ImageVec1 \\ ImageVec2 \\ . \\ . \\ ImageVec20 \end{pmatrix}$$

which gives us a starting point for the PCA analysis. Once PCA is performed, we have original data in terms of the eigen vectors found from the covariance matrix. Why is this useful? Say we want to do facial recognition, and so our original images were of peoples faces. Then, the problem is, given a new image, whose face from the

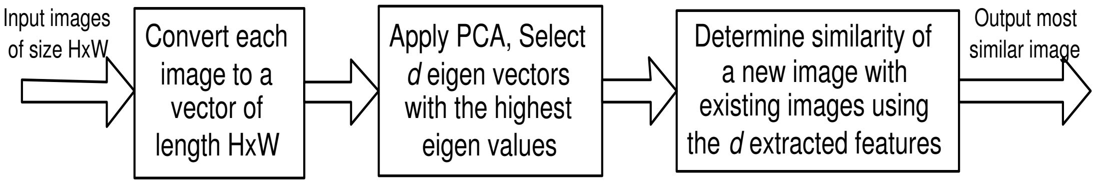
Input images of size HxW
Convert each image to a vector of length HxW
Apply PCA, Select *d* eigen vectors with the highest eigen values
Determine similarity of a new image with existing images using the *d* extracted features
Output most similar image

Figure 5. Procedure for face recognition

original set is it? The way this is done in computer vision is to measure the difference between the new image and the original images, not along the original axes, but along the new axes derived from the PCA analysis. It turns out that these new axes work much better for recognizing faces, because the PCA analysis has extracted these new axes *based on their ability to capture the variability among the images*. In a way, the PCA analysis is able to identify the statistical patterns in the data. Since all the vectors have HXW dimensions, we will get HXW eigen vectors. In practice, one is able to leave out most of the less significant eigen vectors, and the recognition still performs well.

PCA for Image Compression: Using PCA for image compression is also known as the Hotelling, or Karhunen and Loeve (KL) transform. If there are 20 images, each with HXW pixels, one can form HXW vectors, each with 20 dimensions. Each vector consists of all the intensity values from the same pixel from each picture. Notice that this is different from the previous example. By performing PCA on this one gets 20 eigenvectors because each vector is 20-dimensional. To compress the data, one then chooses to transform the data only using, say 5 of the eigenvectors. This gives a final dataset with only 5 dimensions which has saved 3/4 of the space. However, when the original data is reproduced, the images have lost some of the information. This compression technique is said to be lossy because the decompressed image is not exactly the same as the original.

Applications of PCA to Microarray Gene Expression data Each data point produced by a DNA microarray hybridization experiment represents the ratio of expression levels of a particular gene under two different experimental conditions. The result, from an experiment with n genes on a single chip, is a series of n expression-level ratios. Typically, the numerator of each ratio is the expression level of the gene in the varying condition of interest, whereas the denominator is the expression level of the gene in some reference condition. The data from a series of m such experiments may be represented as a gene expression matrix, in which each of the n rows consists of an m-element expression vector for a single gene. The expression measurement is positive if the gene is induced (turned up) with respect to the reference state and negative if it is repressed (turned down).

A PCA analysis of DNA microarray data can consider the genes as variables or the experiments as variables or both. When genes are variables, the analysis creates a set of "principal gene components" that indicate the features of genes that best explain the experimental responses they produce. When experiments are the variables, the analysis creates a set of "principal experiment components" that indicate the features of the experimental conditions that best explain the gene behaviors they elicit. When both experiments and genes are analyzed together, there is a combination of these affects. In [41] the authors considered the experiments as variables. They applied PCA to the publicly released yeast sporulation data set [5]. They found that most of the variance ($>$ 90%) in the sporulation data set is contained in the first two principal components allowing most of the information to be visualized in two dimensions.

Application of SVD to Document Indexing Regular keyword searches approach a document collection with a kind of accountant mentality: a document contains a given word or it does not, with no middle ground. We create a result set by looking through each document in turn for certain keywords and phrases, tossing aside any document that does not contain them, and ordering the rest based on some ranking system.

Latent semantic indexing (LSI) adds an important step to the document indexing process [38]. In addition to recording which keywords a document contains, the method examines the document collection as a whole, to see which other documents contain some of those same words. LSI considers documents that have many words in common to be semantically close, and ones with few words in common to be semantically distant. This simple method correlates surprisingly well with how a human being, looking at content, might classify a document collection. When one searches an LSI-indexed database, the search engine looks at similarity values it has calculated for every content word, and returns the documents that it thinks best fit the query. Because two documents may be semantically very close even if they do not share a particular keyword, LSI does not require an exact match to return useful results. Where a plain keyword search will fail if there is no exact match, LSI will often return relevant documents that do not contain the keyword at all. For example, searching for 'Saddam Hussain' can return documents on Iraq which has no mention of 'Saddam Hussain' in it.

The first step in doing LSI is culling all those extraneous words from a document, leaving only *content words* likely to have semantic meaning. Using this list of content words and documents, we can now generate a term-document matrix. This is a very large grid, with documents listed along the horizontal axis, and content words along the vertical axis. For each content word in our list, we go across the appropriate row and put an 'X' in the column for any document where that word appears. If the word does not appear, we leave that column blank. The key step in LSI is decomposing this matrix using SVD. LSI works by projecting this large, multi-dimensional space down into a smaller number of dimensions. A typical term space might have tens of thousands of dimensions, and be projected down into fewer than 150. In this reduction, information is lost, and content words are superimposed on one another. What we are losing is noise from our original term-document matrix, revealing similarities that were latent in the document collection. Similar things become more similar, while dissimilar things remain distinct. This reductive mapping is what gives LSI its seemingly intelligent behavior of being able to correlate semantically related terms. We are really exploiting a property of natural language, namely that words with similar meaning tend to occur together.

4.2 Uses of Feature Selection Methods

Image Retrieval Feature selection is applied in [46] to content based image retrieval. Recent years have seen a rapid increase of the size and amount of image collections from both civilian and military equipment. However, we cannot access or make use of the information unless it is organized so as to allow efficient browsing, searching and retrieval. Content based image retrieval [43] is proposed to efficiently handle large scale image collections. Instead of being manually annotated by text based keywords, images would be indexed by their own visual contents (features), such as color, texture, and shape. One of the biggest problems to make content based image retrieval truly scalable to large sized image collections is still the curse of dimensionality [20]. As suggested in [43], the dimensionality of the feature space is normally of the order of 10^2 . Dimensionality reduction is a promising approach to solve this problem. The image retrieval system proposed in [46] performs feature selection, and these features are then used to index images for efficient retrieval.

Customer Relationship Management (CRM) A case of feature selection is presented in [36] for customer relationship management. In this context, each customer means a big revenue and the loss of one will likely trigger a significant segment to defect, it is imperative to have a team of highly experienced experts monitor each customer's intention and movement based on massively collected data. A set of key indicators, proven useful in predicting potential defectors, are used by the CRM team. The problem is that it is difficult to find new indicators describing the dynamically changing business environment among many possible features. The machine recorded data is simply too enormous for any human expert to browse and obtain any insight from. Feature selection is employed to search for possible new indicators. They are later presented to experts for scrutiny. This approach considerably improves the team's efficiency in finding new changing indicators.

Intrusion Detection As network based computer systems play increasingly vital roles in modern society, they have become the targets of our enemies and criminals. The security of a computer system is compromised when an intrusion takes place. Intrusion detection is often used as one way to protect computer systems. In [28], Lee, Stolfo, and Mok proposed a systematic data mining framework for analyzing audit data and constructing intrusion detection models. Under this framework, a large amount of audit data is first analyzed using data mining algorithms in order to obtain the frequent activity patterns. These patterns are then used to guide the selection of system features as well as for the construction of additional temporal and statistical features for another phase of automated learning. Classifiers based on these selected features are then inductively learned using the appropriately formatted audit data. These classifiers can be used as intrusion detection models since they can classify whether an observed system activity is “legitimate” or “intrusive”. Feature selection plays an important role in building such classification models for intrusion detection.

Genomic Analysis Structural and functional data from analysis of the human genome has increased many folds in recent years, presenting enormous opportunities and challenges for data mining. In particular, gene expression microarray is a rapidly maturing technology that provides the opportunity to assay the expression levels of thousands or tens of thousands of genes in a single experiment. These assays provide the input to a wide variety of data mining tasks, including classification and clustering. However, the number of instances in these experiments is often severely limited. In [51], for example, Xing et al used a case involving only 38 training data points in a 7130 dimensional space to exemplify the above situation which is becoming increasingly common in molecular biology applications. In this extreme case of very few observations on a large number of features, Xing et al investigated the possible use of feature selection on a microarray classification problem. All the classifiers tested in the experiments performed significantly better in the reduced feature space than in the full feature space.

Text Categorization is the problem of automatically assigning predefined categories to free text documents [29, 37]. This problem is of great practical importance given the massive volume of online text available through the World Wide Web, Emails, and digital libraries. A major characteristic, or difficulty of text categorization problems is the high dimensionality of the feature space. This is prohibitively high for many mining algorithms. Therefore, it is highly desirable to reduce the original feature space without sacrificing categorization accuracy. In [54], different feature selection methods are evaluated and compared to reduce high dimensional space in text categorization problems. It is reported that the methods under evaluation can effectively remove 50% - 90% of the terms while maintaining the categorization accuracy.

5. Conclusions and Future Directions

As computers become increasingly powerful, many applications can produce massive data of high dimensionality. Dimensionality reduction is an efficient way of dealing data with high dimensionality. The purpose is to reduce the data so that computational load decreases and patterns of better quality can be extracted by pattern recognition and data mining algorithms. In this article, we described the concepts of feature extraction and feature selection, and briefly introduced some representative methods. We then presented in brief some cases of dimensionality reduction to illustrate its application to many problem domains. The need of dimensionality reduction techniques presents new challenges, and novel methods are expected to be developed.

One future research direction is to extend these techniques to different application areas such as microarray gene expression data. Typically microarray data has many genes but very less number of sample tests thus suffering from the curse of dimensionality. Another research direction is to select tuples and combine it with dimensionality reduction method. Usually researchers have been performing dimensionality reduction and tuple selection separately. Some other research directions include Kernel PCA, probabilistic PCA, and independent component analysis.

References

[1] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *Proceedings of ACM SIGMOD Conference on Management of Data*, pages 61–72, 1999.

[2] R Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of ACM SIGMOD Conference on Management of Data*, 1998.

[3] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, 1961.

[4] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[5] S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P.O. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998.

[6] M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering – A filter solution. In *Proceedings of IEEE International Conference on Data Mining, (ICDM)*, 2002.

[7] M. Dash and H. Liu. Feature selection for classification. *International Journal of Intelligent Data Analysis*, 1(3), 1997.

[8] M. Dash and H. Liu. Feature selection for clustering. In *Proceedings of Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining, (PAKDD)*, 2000.

[9] S De Backer, A. Naud, and P. Scheunders. Non-linear dimensionality reduction techniques for unsupervised feature extraction. *Pattern Recognition Letters*, 19:711–720, 1998.

[10] M. Devaney and A. Ram. Efficient feature selection in conceptual clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 92–97, 1997.

[11] J. Doak. An evaluation of feature selection methods and their application to computer security. Technical report, Davis CA: University of California, Department of Computer Science, 1992.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley Interscience, 2001.

[13] G. H. Dunteman. *Principal Components Analysis*. Sage Publications, 1989.

[14] J. G. Dy and C. E. Brodley. Visualization and interactive feature selection for unsupervised data. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 360–364, 2000.

[15] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, Aug 2004.

[16] U.M. Fayyad and R. Uthurusamy. Evolving data mining into solutions for insights. *Communications of the Association for Computing Machinery*, 45(8):28 – 31, August 2002.

[17] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[18] D. Foley and J. W. Sammon. An optimal set of discriminant vectors. *IEEE Transactions on Computers*, 24 (5):281–289, 1975.

[19] J.H. Friedman. On bias, variance, 0/1 - loss, and the curse-of-dimensionality. Technical report, Stanford University, 1996.

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[21] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[22] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In W.W. Cohen and Hirsh H., editors, *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129, New Brunswick, N.J., 1994. Rutgers University.

[23] I. T. Joliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[24] J. Kittler. *Feature selection and extraction*, pages 59 – 83. Academic Press, Orlando, 1986.

[25] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. Som pak: The self-organizing map program package, 1996.

[26] I. Kononenko. Estimating attributes : Analysis and extension of RELIEF. In *Proceedings of European Conference on Machine Learning (ECML)*, pages 171–182, 1994.

[27] J.B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, Beverly Hills, CA, 1978.

[28] W. Lee, S. J. Stolfo, and K. W. Mok. Adaptive intrusion detection: A data mining approach. *AI Review*, 14(6):533 – 567, 2000.

[29] E. Leopold and Kindermann J. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46:423–444, 2002.

[30] H. Liu and H. Motoda, editors. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Boston: Kluwer Academic Publishers, 1998.

[31] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers, 1998.

[32] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In L. Saitta, editor, *Proceedings of International Conference on Machine Learning (ICML-96), July 3-6, 1996*, pages 319–327, Bari, Italy, 1996. San Francisco: Morgan Kaufmann Publishers, CA.

[33] G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley and Sons, New York, 1992.

[34] A. N. Mucciardi and E. E. Gose. A comparison of seven techniques for choosing subsets of pattern recognition. *IEEE Transactions on Computers*, C-20:1023–1031, September 1971.

[35] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature selecting. In *IEEE Transactions on Computers, C-26(9)*, 1977.

[36] K.S. Ng and H. Liu. Customer retention via data mining. *AI Review*, 14(6):569 – 590, 2000.

[37] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.

[38] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the ACM Conference on Principles of Database Systems (PODS)*, 1998.

[39] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[40] J. R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[41] S. Raychaudhuri, J. M. Stuart, and R. B. Altman. Principal components analysis to summarize microarray experiments : application to sporulation time series. In *Proceedings of Pacific Symposium on Biocomputing*, pages 455–466, 2000.

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

[43] Y. Rui, T. S. Huang, and S. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[44] J. W. Sammon. A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18 (5):401–409, 1969.

[45] J. C. Schlimmer. Efficiently inducing determinations : a complete and systematic search algorithm that uses optimal pruning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 284–290, 1993.

[46] D. L. Swets and J. J. Weng. Efficient content-based image retrieval using automatic feature selection. In *IEEE International Symposium On Computer Vision*, pages 85–90, 1995.

[47] L. Talavera. Feature selection as a preprocessing step for hierarchical clustering. In *Proceedings of International Conference on Machine Learning (ICML)*, 1999.

[48] L. Talavera. Feature selection and incremental learning of probabilistic concept hierarchies. In *Proceedings of International Conference on Machine Learning (ICML)*, 2000.

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

[50] S. Vaithyanathan and B. Dom. Model selection in unsupervised learning with applications to document clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 433–443, 1999.

[51] E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference On Machine Learning*, 2001.

[52] Ng. A. Y. Preventing overfitting of crossvalidation data. In *Proceedings of Fourteenth International Conference on Machine Learning*, pages 245–253, 1997.

[53] Shisong Yang and Chih-Cheng Hung. Image texture classification using datagrams and characteristic views. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 22–26, 2003.

[54] Y. Yang and J. O. Pederson. A comparative study on feature selection in text categorization. In *Proceedings of Fourteenth International Conference on Machine Learning*, pages 412–420, 1997.

[55] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, Oct 2004.