A Novel Scalable Algorithm for Supervised Subspace Learning

Jun Yan[1], Ning Liu[1], Benyu Zhang[1], Qiang Yang[2], Shuicheng Yan[3], Zheng Chen[1]
[1]*Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, P.R. China*
{junyan, ningl, byzhang, zhengc}@microsoft.com
[2]*Department of Computer Science, Hong Kong University of Science and Technology*
qyang@cs.ust.hk
[3]*ECE department, University of Illinois at Urbana Champaign, USA*
scyan@ifp.uiuc.edu

Abstract

Subspace learning approaches aim to discover important statistical distribution on lower dimensions for high dimensional data. Methods such as Principal Component Analysis (PCA) do not make use of the class information, and Linear Discriminant Analysis (LDA) could not be performed efficiently in a scalable way. In this paper, we propose a novel highly scalable supervised subspace learning algorithm called as Supervised Kampong Measure (SKM). It assigns data points as close as possible to their corresponding class mean, simultaneously assigns data points to be as far as possible from the other class means in the transformed lower dimensional subspace. Theoretical derivation shows that our algorithm is not limited by the number of classes or the singularity problem faced by LDA. Furthermore, our algorithm can be executed in an incremental manner in which learning is done in an online fashion as data streams are received. Experimental results on several datasets, including a very large text data set RCV1, show the outstanding performance of our proposed algorithm on classification problems as compared to PCA, LDA and a popular feature selection approach, Information Gain (IG).

1. Introduction

In the last decade, machine learning and data mining research has witnessed a growing interest in subspace learning [7] and its applications, such as Web document classification [11], face recognition [16] and data clustering [1, 12]. Among various subspace learning approaches, linear algorithms are of great interesting due to their efficiency and effectiveness. Principal Component Analysis (PCA) [2, 13] and Linear Discriminant Analysis (LDA) [4] are two of the most widely used traditional linear subspace learning algorithms.

Principal Component Analysis (PCA), which is an unsupervised algorithm, aims at finding out the geometrical structure of the data set and projecting the data along the directions with maximal variances. However, it discards the class information which is significant for classification tasks. Linear Discriminant Analysis (LDA), which is a supervised algorithm also called as Fisher Discriminant Analysis (FDA), is a traditional supervised subspace learning algorithm. It searches for the projection on which the data points of different classes are far from each other and, at the same time, the data points of the same class are close to each other. Nevertheless, the available subspace dimension in LDA is limited by the number of classes, and the singularity problem limits the application of LDA [10].

On the other hand, both classical PCA and LDA are batch algorithms, which mean that the training data must be available in advance. As a result, they cannot satisfy the requirements of online applications on data stream [3, 6, 14]. Furthermore, when the dimensionality of the features is high, the computational complexity and the storage requirement grow dramatically. Thus in this paper we propose a novel algorithm which can satisfy: (1), it is supervised which is more suitable to classification problems than the common used unsupervised one, PCA; (2), it has no calculation limitations such as the class number limitation and singularity problem faced by LDA, in addition it can give the same or better performance than LDA for real classification tasks; (3), the computational complexity is lower than the traditional matrix decomposition based algorithms PCA and LDA; (4), it can deal with data streams in a scalable way.

We name the novel proposed supervised subspace learning approach with incremental learning as Supervised Kampong Measure (SKM). Note SKM is just a name which has no physical meaning. Intuitively in the history of human being, to avoid hurts of animals, the residents of the same remote antiquity Kampong always live together as close as possible, on the other hand, to avoid war and get enough food, the residents of different kampongs always live as far as possible. Similar to the custom of kampong residents, we design SKM aim at assigning data points as close as possible to their corresponding class means, and simultaneously, assigning data points as far as possible to the other class means in the low dimensional subspace.

As compared to the classical LDA, the optimization of our proposed measure does not depend on the number of

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

training data's classes; and it has no singularity problem. Moreover, it outperforms PCA in classification problems since PCA discards the valuable class information. Our experimental results on various data sets show that this algorithm is effective for classification problems compared to PCA or LDA. Particularly, experiments performed on Reuters Corpus Volume 1 (RCV1) [8], whose dimension is about 300'000 and the samples number is more than 800'000, demonstrate the scalable property of SKM on a very large scale dataset. We take unsupervised PCA as a baseline of SKM on the large scale data. In addition, based on the comparative study of [17], supervised IG is the best one among several feature selection algorithms for text categorization problems. Thus we also utilize IG as baseline of SKM. Since it is very hard to perform LDA on such a large dataset, we only conduct LDA on general datasets. This large scale experiment shows that the F1 value of our algorithm in a 3-dimensional subspace could outperform PCA and IG in even 500-dimensional subspaces.

The rest of this paper is organized as follows. In Section 2, we introduce some necessary background knowledge on subspace learning, such as the PCA and LDA algorithms. In Section 3, we will give the formal problem statement. Following that, we present the derivation of our proposed approach and its incremental learning algorithm in Section 4. In Section 5, we demonstrate the experimental results on the synthetic datasets and the real data. Conclusion of this paper is given in Section 6. Some detailed proof could be found in the appendix.

2. Background Knowledge

Linear subspace learning approaches are widely used in real tasks such as Web document classification and face recognition nowadays. It aims at finding a projection matrix which could efficiently project the data from the original high-dimensional feature space to a much lower dimensional representation under a particular criterion. Different criterion will yield different subspace learning algorithm with different properties. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two most widely used linear subspace learning approaches.

2.1. Principal Component Analysis

Suppose that the sample data points $u(1), u(2), ..., u(N)$ are d-dimensional column vectors, where N is the number of samples, and that U is the sample matrix with $u(i)$ as its i^{th} column. PCA aims to find a subspace whose basis vectors correspond to the directions with maximal variances. It projects the original data into a p-dimensional ($p << d$) subspace. The new p-dimensional feature vector can be computed as $y = W^T u$, where W is the projection matrix and its column vectors correspond to the p leading eigen-vectors of the centralized covariance matrix $C = UU^T$. Here uppercase T stands for transpose of matrix.

PCA minimizes the reconstruction error in the sense of least square error, and finds out the most representative features. The objective function of PCA is,

$$J(W) = tr\{W^T C W\}.$$

In other words, the projection matrix W is achieved by maximizing the objective function $J(W)$.

Moreover, PCA is in fact a scalable algorithm since it has effective incremental learning algorithm [2] which could process large scale streaming data. However, it ignores the class label information which is very valuable for general classification tasks.

2.2. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA), also called Fisher Discriminant Analysis (FDA), was proposed to pursue a low dimensional subspace that can best discriminate the samples from different classes. Suppose $W \in R^{d \times p}$ is the linear projection matrix; LDA aims to maximize the so-called Fisher criterion,

$$J(W) = \frac{tr\{W^T S_b W\}}{tr\{W^T S_w W\}},$$

Where

$$S_b = \sum_{i=1}^{c} p_i (m_i - m)(m_i - m)^T,$$

$$S_w = \sum_{i=1}^{c} p_i E(u_i - m_i)(u_i - m_i)^T$$

are called the Inter-class scatter matrix and the Intra-class scatter matrix respectively, where c is the number of classes, m is the mean of all samples, m_i is the mean of the samples belonging to class i and p_i is the prior probability for a sample belonging to class i. The projection matrix W can be obtained by maximizing the objective function $J(W)$. Through simple mathematical derivation, it is to solve the following generalized eigen-vector decomposition problem:

$$S_b w = \lambda S_w w.$$

LDA explicitly utilizes the class label information of the samples and is a supervised algorithm. There are at most c-1 nonzero eigen-values, so the upper bound of p is c-1; and at least $d + c$ sample data is required to make it possible that S_w is not singular. These limit the application of LDA. Furthermore, it is difficult for LDA

to handle large-sized datasets when the feature-space dimension is high. For example, as in the Reuters Corpus Volume 1, the feature dimension is about 300,000 where it is impossible to conduct the generalized eigen-vector decomposition on a computer with moderate configuration.

Figure 1 shows a simple example of different subspace learning approaches on a group of synthetic data. The stars and triangles are two dimensional data points belong to two different classes. The straight line is the one dimensional subspace found by PCA and the broken line is the one dimensional subspace found by LDA. It is obvious that if we project the data into the subspace calculated by LDA, they will be separated easily. However, if we project the data into the subspace calculated by PCA, the samples of two classes will be mixed together. As a conclusion, the unsupervised PCA project the data based on the overall distribution of dataset while supervised LDA project the data based on the class distribution of dataset. This is one of the reasons why supervised subspace learning approaches are always better than unsupervised ones on some classification tasks.

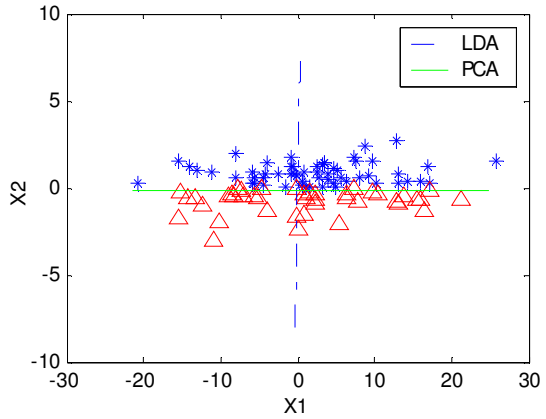

Figure 1. A simple example of subspace calculated by different subspace learning approaches.

3. Problem Formulation

As demonstrated in the previous section, subspace learning approaches attempt to find a projection matrix which could efficiently project the data from the original feature space to a target representation under a certain optimization criterion. In mathematical terms, suppose that we are given N training data $u(i) \in R^d$, $i = 1,2,...N$ where the class label of $u(i)$ is l_i, $l_i \in \{1,2,\ldots,c\}$ and c is the number of classes. The mean of class i is m_i. The problem then is to give an objective function $J(W)$ which satisfies the properties listed below. We can then solve the projection matrix $W \in R^{d \times p}$ ($p << d$) by optimizing this objective function $J(W)$ in order to produce the subspace. The properties we wish the objective function $J(W)$ to satisfy are:

- It should be optimal for classification tasks. In other words, it should outperform at least PCA and at the same time, perform the same or better than LDA for classification problems;
- This criterion should be applicable. In other words, the dimension of subspace should not be limited by the class size. Furthermore, the computation should not be constrained by the singularity problems;
- The optimization of this criterion should be scalable, i.e. suitable for incremental learning with much lower complexity than batch calculation. In other words, it should support both batch as well as online learning.

4. Supervised Kampong Measure

Intuitively in the history of human being, to avoid hurts of animals, the residents of the same remote antiquity Kampong always live together as close as possible, on the other hand, to avoid war and get enough food, the residents of different kampongs always live as far as possible. We are motivated to make use of a similar idea in subspace learning. This is the reason why we call our criterion SKM. We measure the distance between a data point and all the class centers for subspace learning. In other words, we wish to make the distance between a data point and its corresponding class center as short as possible in the extracted low dimension subspace, at the same time, we wish to make the distance between data points and the inhomogeneous class centers as far as possible.

Figure 2 shows the intuitive motivation of the SKM algorithm for subspace learning. For a high dimensional data point that belongs to class 2, suppose P1, P2, P3 are three possible positions after projected to the 2 dimension subspace. Under our motivation, possible position 2 (P2) is better than possible position 1 (P1) since it is closer to its own class mean. Moreover, P3 is better than P2 due to the reason that P3 is one of the closest positions to its own class mean among the three possible positions, and is the furthest to the other class means.

4.1. Derivation of the Criterion

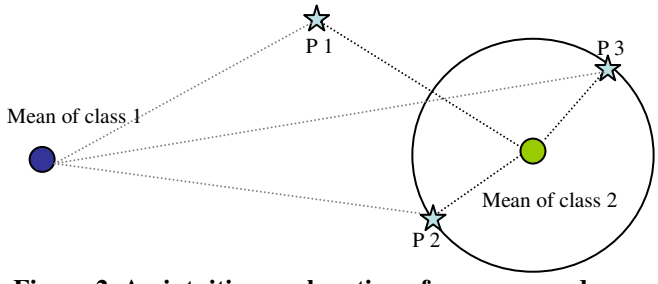

Figure 2. An intuition explanation of our proposed criterion.

Using the same symbols defined in section 2, all data points are *d*-dimensional column vectors in Euclidean Space, thus the distance between the i^{th} sample data and the mean of class j could be represented by,

$$dis(u(i), m_j) = \left\| u(i) - m_j \right\|_2 = \sqrt{(u(i) - m_j)^T (u(i) - m_j)} \quad (1)$$

Through algebra transformation, it is easy to prove that,

$$(u(i) - m_j)^T (u(i) - m_j) = tr\{(u(i) - m_j)(u(i) - m_j)^T\} \quad (2)$$

where $tr\{\cdot\}$ is the trace of a given matrix. In other words, we have

$$tr\{(u(i) - m_j)(u(i) - m_j)^T\} = dis(u(i), m_j)^2 \quad (3)$$

Thus we can use the trace of scatter matrix $(u(i) - m_j)(u(i) - m_j)^T$ to measure the distance between the i^{th} sample data and the mean of class j.

Suppose that $W \in R^{d \times p}$ is the linear projection matrix to be found by our algorithm through which a new *p*-dimensional feature vector can be computed as $y = W^T u$. The distance between the i^{th} sample data and the mean of class j in the low dimensional subspace then could be denoted by,

$$tr\{W^T (u(i) - m_j)(u(i) - m_j)^T W\} \quad (4)$$

From the supervised learning perspective, different class has different prior probability. Intuitively, the stronger a kampong is, to avoid conflict and hurt, the farther other kampong's residents should live away from it. In terms of subspace learning, the larger prior probability of a class is, the larger distance between its class mean and a data point of other class should be. This leads to a weighted form of formula (4). Suppose that l_i stands for the class label of $u(i)$ and $e_{l_i, j}$, $j = 1, 2, ...c$ are the weights of different class centers with a determined data point (we discuss how to set these weights in the next section). Then the measurement of distance between $u(i)$ and m_j is:

$$tr\{W^T e_{l_i, j} (u(i) - m_j)(u(i) - m_j)^T W\} \quad (5)$$

In order to make a people live far away from all other kampongs except for his own, we should maximize the distance between a data point $u(i)$ and all other class centers in the low dimensional subspace, i.e. we should maximize,

$$tr\{W^T (\sum_{j=1, j \neq l_i}^{c} e_{l_i, j} (u(i) - m_j)(u(i) - m_j)^T) W\} \quad (6)$$

On the other hand, in order to minimize the distance between a data point and its corresponding class centers after being projected on a low dimensional subspace, we minimize,

$$tr\{W^T e_{l_i, l_i} (u_{l_i}(i) - m_{l_i})(u_{l_i}(i) - m_{l_i})^T W\} \quad (7)$$

To get a unified formula, we define a δ-function as,

$$\delta(l_i, j) = \begin{cases} -1 & l_i = j \\ 1 & l_i \neq j \end{cases} \quad (8)$$

Then we can combine formular (6) and (7) as,

$$tr\{W^T (\sum_{j=1}^{c} \delta(l_i, j) e_{l_i, j} (u(i) - m_j)(u(i) - m_j)^T) W\} \quad (9)$$

Note that if formula (9) gets its maximum value, then our original motivation i.e. assigning data points as close as possible to their corresponding class means, meanwhile, assigning data points as far as possible to the inhomogeneous class means is satisfied for a given data point $u(i)$. And then for all the sample data, the full criterion could be write as,

$$\begin{aligned} J(W) &= tr\{W^T (E\{\sum_{j=1}^{c} \delta(l_i, j) e_{l_i, j} (u(i) - m_j)(u(i) - m_j)^T\}) W\} \\ &= tr\{W^T S_c W\} \end{aligned} \quad (10)$$

where $E\{\cdot\}$ is the expectation of a random variable and

$$S_c = E\{\sum_{j=1}^{c} \delta(l_i, j) e_{l_i, j} (u(i) - m_j)(u(i) - m_j)^T\} \quad (11)$$

Based on the above discussion, the SKM for subspace-learning is transformed into an optimization problem. In the equation (11), we exercised freedom to multiply W with some nonzero constant. Thus, we additionally require that W consists of unit vectors, i.e. $W = [w_1, w_2, \text{L}\ w_p]$ and $w_k^T w_k = 1$. In addition, we require the projection matrix W to be orthogonal matrix in Euclidean space, i.e. $w_k^T w_l = 0$ if $k \neq l$. Then the optimization problem of the proposed objective function (10) is translated to the following constraint optimization problem:

$$W^* = \arg\max_{W^T W = I} J(W) = \arg\max_{W^T W = I} tr\{W^T S_c W\} \quad (12)$$

where *I* is the identity matrix. The problem can be restated as,

$$\begin{aligned} &\arg\max \sum_{k=1}^{p} w_k^T S_c w_k, \\ &\text{subject to } w_k^T w_l = \begin{cases} 1 & k = l \\ 0 & k \neq l \end{cases}, \\ &k = 1, 2, \ldots, p. \\ &l = 1, 2, \ldots, p. \end{aligned}$$

4.2. Design of the Weights

In this section we consider the design of weights $e_{l_i,j}$ in objective function (10). Note $j \neq l_i$ indicates that the sample $u(i)$ does not belong to class j, in which situation we want to separate $u(i)$ from samples of class j in the p-dimensional subspace. As demonstrated in section 4.1, the more samples in class j, the larger the weights are needed for class j, i.e. the longer the distance is needed to separate them. Alternatively, when $j = l_i$, we wish to associate $u(i)$ with samples of class j in the p-dimensional subspace. Intuitively from the global perspective, if a kampong has a few residents, then everyone is very important to this society. All people should live very close for safety. However if a kampong has a lot of residents, people can live a little far away from each other since the loss of a single person will not affect the kampong much. Thus the more samples in class j, the smaller the weights should be for class j.

The analysis above tells us that the weights must satisfy the following constraints: (a), if $u(i)$ does not belong to class j, the weight $e_{l_i,j}$ should be monotone increasing with the prior probability of class j; (b), in contrast, if $u(i)$ belongs to class j, the weight $e_{l_i,j}$ should be monotone decreasing with the prior probability of class j; (c), the third is a common constraint that all weights must satisfy $e_{l_i,j} \geq 0$ and $\sum_{j=1}^{c} e_{l_i,j} = 1$; (d), finally, a people live together with his own kampong is more important than live far away from other kampongs. In other words, the distance between a data point and its corresponding class mean should be more important than the distance between the data point and all the other class centers, i.e. $e_{l_i,l_i} \geq \sum_{j \neq l_i} e_{l_i,j}$.

It is clear that we can design various weights to satisfy these constraints. As an example, we assign the weights which can satisfy (a) ~ (d) by Theorem 1.

Theorem 1, the weights

$$e_{l_i,j} = \begin{cases} (a - p_j)/M & j = l_i \\ p_j / M & j \neq l_i \end{cases}, \quad (13)$$

satisfies all the constraints (a) ~ (d) discussed above when $a \geq 1$, where M is a positive real const used to normalize the weights (a simple proof is given in appendix)

Without loss of generality, we could use

$$e_{l_i,j} = \begin{cases} (a - p_j) & j = l_i \\ p_j & j \neq l_i \end{cases}, \quad (14)$$

to represent the weights in our problem. This is due to the reason that multiplying by a constant will never affect the projection matrix, mathematically

$$\arg\max_{W^T W = I} tr\{W^T S_c W\} = \arg\max_{W^T W = I} tr\{W^T (M S_c) W\}. \quad (15)$$

For convenience, we combine the weights and the δ-function as,

$$q(l_i, j) = \begin{cases} -(a - p_j) & l_i = j \\ p_j & l_i \neq j \end{cases}, \quad (16)$$

and then the objective function can be rewritten as,

$$\begin{aligned} J(W) &= tr\{W^T (E\{\sum_{j=1}^{C} q(l_i, j)(u(i) - m_j)(u(i) - m_j)^T\}) W\} \\ &= tr\{W^T S_c W\} \end{aligned} \quad (17)$$

4.3. Algorithm Analysis

Note our algorithm is an optimization problem, $\arg\max \sum_{k=1}^{p} w_k^T S_c w_k$ subject to some constraints. By introducing a Lagrangian function below,

$$L(w_k, \lambda_k) = \sum_{k=1}^{p} w_k^T S_c w_k - \lambda_k (w_k^T w_k - 1) \quad (18)$$

where λ_k are the Lagrange multipliers. At the saddle point, the derivatives of L must vanish, leading to $S_c w_k = \lambda_k w_k$. Thus, the columns of W are eigen-vectors of the criterion matrix S_c. Therefore, $J(W)$ is maximized when W is composed of the first p leading eigen-vectors of S_c.

Compared to the classical LDA algorithms, our proposed algorithm has no singularity problem since it need not to calculate the inversing of a matrix. In addition, our proposed algorithm is not limited by the number of classes.

4.4. Incremental Learning

The above eigen-system could be solved by the classical Singular Value Decomposition (SVD), but the computational complexity of SVD is $O(m^3)$, where m is the smaller value between the sample number and the data dimension. However, in real applications, we allow data to be streamed from a data source, such that the data are incrementally received. Furthermore, when the dimension of the data set is high, both the computational and storage costs grow fast. Thus, an incremental method is highly desired to compute an adaptive subspace when the data arrive sequentially.

Assume that the sample sequence is presented as $\{u(n)\}$, where n=1, 2…. Our algorithm aims to maximize the criterion $J(W) = W^T S_c W$. Denote p as the dimension of the transformed data, i.e. the final subspace dimension. The criterion scatter matrix of step n after learning from the first n samples can be written as,

$$S_c(n) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} q(l_i, j)(u(i) - m_j(i))(u(i) - m_j(i))^T \quad (19)$$

where $m_j(i)$ is the center of class j at step i.

The general eigen-function is $A\mu = \lambda\mu$, where eigen-value of A is λ and the corresponding eigen-vector is μ. By replacing the matrix A with $S_c(n)$, we can obtain an approximate iterative eigen-vector computation formulation with $v = \lambda\mu$:

$$v(n) = S_c(n)\mu(n) \quad (20)$$
$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} q(l_i, j)(u(i) - m_j(i))(u(i) - m_j(i))^T \mu(i) .$$

Then the eigen-vector can be directly computed as $\mu = v/\|v\|$. For iterative calculation, let $\mu(i) = v(i-1)/\|v(i-1)\|$, we have the following incremental formulation:

$$v(n) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} q(l_i, j)(u(i) - m_j(i))(u(i) - m_j(i))^T \frac{v(i-1)}{\|v(i-1)\|} \quad (21)$$

Through simple algebra derivation, we can get

$$v(n) = \frac{n-1}{n}v(n-1) + \quad (22)$$
$$\frac{1}{n}\sum_{j=1}^{c} q(l_n, j)(u(n) - m_j(n))(u(n) - m_j(n))^T \frac{v(n-1)}{\|v(i-1)\|}$$

For initialization, we set $u(1)$ as the first sample.

Notice that eigen vectors are orthogonal to each other. So, it helps to generate "observations" only in a complementary space for computation of the higher order eigen-vectors. To compute the $(j+1)^{th}$ eigen-vector, we first subtract its projection on the estimated j^{th} eigen-vector from the data,

$$u^{j+1}(n) = u^j(n) - (u^j(n)^T v^j(n))v^j(n)/\|v^j(n)\|\|v^j(n)\|, \quad (23)$$

where $u^1(n) = u(n)$. In this way, the time-consuming orthonormalization is avoided and the orthogonality is always enforced when the convergence is reached, although not exactly so at early stages. A very similar convergence proof of this algorithm could be found in [15]. Then the full algorithm is computing (22) (23) iteratively from the initially value.

The time complexity of incremental SKM to train N input samples is $O(Ncdp)$, where c is the number of classes, d is the dimension of the original data space, and p is the target dimension, which is linear with each factor. Furthermore, when handling each input sample, SKM only need to keep the learned eigen-space and several first-order statistics of the past samples, such as the mean and the counts. Hence, SKM is scalable which can handle large scale and continuous data.

5. Experimental Results

In order to test our algorithm, we conducted three sets of experiments. In the first set, we tested the incremental algorithm on a very high dimensional and large dataset RCV1 whose dimension is about 300,000. In the second experiment, we used a synthetic dataset generated using a normal distribution. The purpose of this experiment is to illustrate the subspaces learned by LDA, PCA and SKM algorithms follow our initial intuition. In the third set of experiments, we applied our method to some UCI datasets [5] to compare the classification performance with other approaches.

5.1. Reuters - Large Scale Text Data

In order to demonstrate the performance of Incremental SKM on a high dimensional and large scale data set, we tested our algorithm on the Reuters Corpus Volume 1 (RCV1). We show the performance of a=1 in this test; we observed that using other values of a resulted in similar performance.

To compare the effectiveness of Incremental SKM algorithm with other subspace learning algorithms, we constructed classification experiments on RCV1. The dimension of each sample data is about 300,000, where each dimension is defined by a keyword.

We chose the data samples with the highest four topic codes (CCAT, ECAT, GCAT, and MCAT) in the "Topic Codes" hierarchy, which contains 789,670 documents. Then we applied a five-fold cross validation on the data. We split them into five equal-sized subsets, where in each experiment four of them are used as the training set and the remaining one is left as the test set. The experimental results reported are the average of the five runs. The detailed experiments following the 5 steps listed below,

- Applying the dimension reduction algorithm on a specific size of training data to learn a subspace;
- Transforming all the training data to the subspace;
- Training SVM classifier by the reduced training data;
- Transforming all the test data to the subspace;
- Evaluate the classification performance, using F1 value, on the transformed test data.

Figure 3 gives the F1 value, which is a common used evaluation measure for text classification, of the incremental algorithm using Support Vector Machine (SVM) as the classifier. In this experiment, the Information gain (IG) and Incremental Principal Component Analysis (IPCA) algorithms are used as baselines.

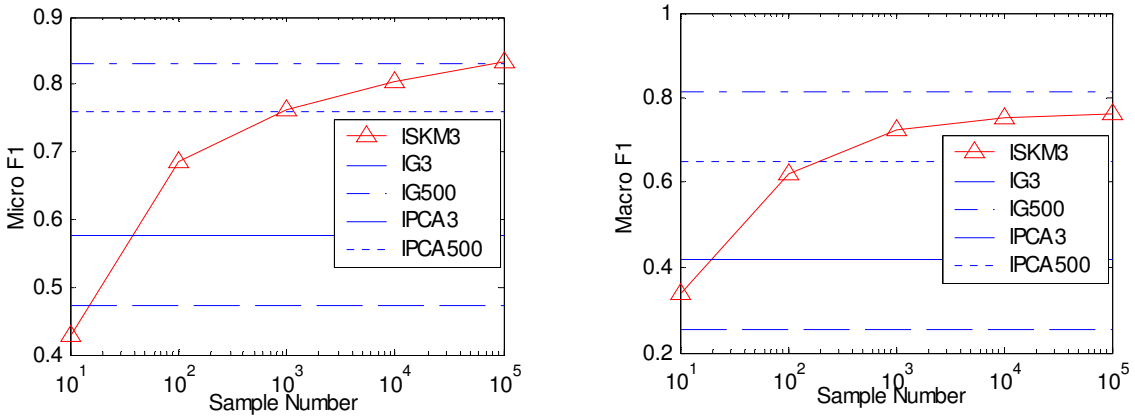

Figure 3. F1 value of incremental SKM

The numbers after the tags in figure 3 denote the subspace dimension. For example "IG500" means the 500-dimension subspace found by IG. The y-axis of Figure 3 are Micro-F1 and Macro-F1 respectively. The larger F1 value is, the better classification performance is achieved. The x-axis is the number of training samples used in our incremental algorithm. Note that we use all the training data when we train the baseline algorithms for dimension reduction.

From this experiment, we can observe that the 3-dimensional subspace calculated by incremental SKM is much better than even the 500-dimensional subspace of Principal Component Analysis. Moreover, for IG, one of the most popular dimensionality reduction approaches for large scale text data, SKM on 3-dimensional subspace outperforms it on the same scale and SKM3 has comparable performance with IG500.

SKM outperforms PCA for classification problems since the former is supervised approach while the latter is unsupervised which ignores the valuable class label information. SKM outperforms IG due to the reason that IG is feature selection approach. It directly selects features from the original data space while SKM use a transformation (projection) matrix to reduce the very high dimension of data. Since the text data are sparse dataset, feature selection approaches always reduce many different data points to the same zero vectors when the reduced dimension is low.

Table 1 gives the comparison of all the different approaches considered by us.

Table 1. Summary of different approach

	A	**B**	**C**	**D**
SKM	YES	YES	YES	YES
PCA	**NO**	YES	YES	YES
LDA	YES	**NO**	YES	**NO**
IG	YES	YES	**NO**	YES

(A), supervised approach which is usually better than unsupervised approaches for classification;

(B), scalable approach which could be used to large scale dataset;

(C), feature extraction approach which could reduce large scale sparse text data to very low dimensional space;

(D), a free approach which is not limited by the data structure. For example, for a large scale dataset with the dimension more than 10'000, if the number of classes is only 3, the subspace dimension by LDA could not beyond 2.

5.2. Synthetic Data Set

In this experiment we show the performance of SKM by generating a 3-dimensional dataset with two classes for intuition. As an example, each class consists of 100 samples following the norm distribution with means (0,0,0) and (5,5,5). Figure 4-(a) shows a scatter plot of the data set. (b), (c), (d) (e) (f) are low dimensional projection of original data by different subspace learning approaches. In (b) we project the data into 1-dimensional subspace by PCA; in (c) and (d) we project the data into 1-dimensional subspace by SKM and LDA respectively; in (e) and (f) we project the data into 2-dimensional subspace by PCA and SKM respectively.

From (b) and (e) we can see that the PCA subspace mixes these two classes of data in both one dimension and two. Though LDA subspace could separate them in figure (d), the dimension of LDA subspace could not beyond one, i.e. we can not project the data into 2-dimensional subspace due to the limitation of class number. (c) and (f) show that our proposed approach could separate these two class in any dimensional subspace for this synthetic dataset.

From this experiment we can see intuitively that PCA is not optimal for classification tasks and mixes the two classes; the subspace dimension of LDA could not beyond one due to its limitation. It is clear that our proposed algorithm outperform PCA and do not limited by the number of classes.

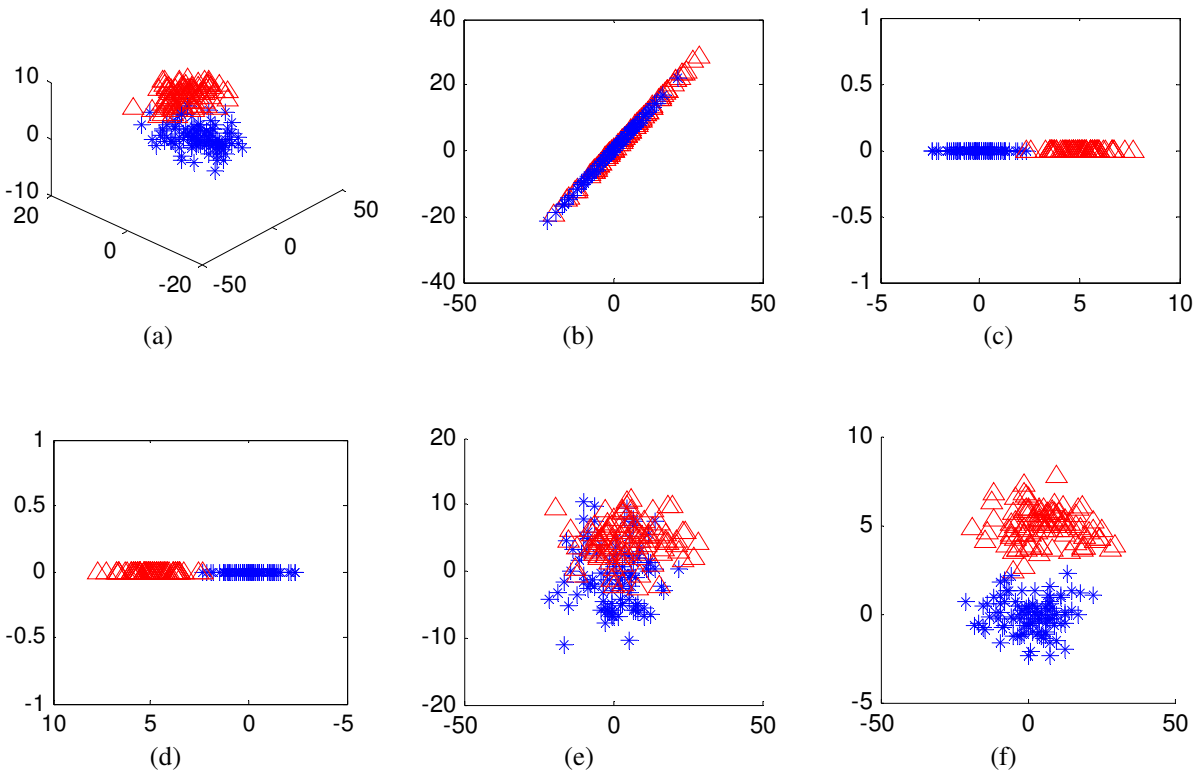

Figure 4. Synthetic data experiment.

(a), original data sets;(b), projected to 1 dimension by PCA; (c), projected to 1 dimension by SKM; (d), projected to 1 dimension by LDA;(e), projected to 2 dimension by PCA; (f), projected to 2 dimension by SKM;

5.3. UCI Data

The UCI machine learning dataset is a repository of databases, domain theories and data generators that are used for the empirical analysis of machine learning algorithms. For each UCI dataset that do not provide training-testing split, we used repeated holdout methods by repeatedly separating them into two folds randomly. SKM, PCA and LDA are then applied to the training data to find the subspace.

The k-nearest neighbor classifier is used to classify these testing data. By using the same classifier, we use classification error rate to evaluate the performance of different subspace learning algorithms for classification tasks. 6 UCI subsets are utilized in this paper for experiments.

(1), "IRIS" has 150 samples with 2 classes. The dimension of data is 4.

(2), "Isolet Spoken Letter" has 6927 samples with 26 classes. The dimension of data is 617.

(3), "Wine" has 178 samples with 3 classes. The dimension of data is 13.

(4), "Letter" has 20000 samples with 26 classes. The dimension of data is 16.

(5), "Monk's Problem" has 556 samples with 2 classes. The dimension of data is 6.

(6), "BUPA" has 345 samples with 6 classes. The dimension of data is 2.

Figure 5 shows the picture of error rate with subspace dimension on these subsets of UCI. It seems that supervised subspace learning approaches outperform unsupervised PCA most of the time (a), (b), (c), (d), (e). However, the subspace dimension was limited for LDA. Moreover, SKM still could outstand even on the dataset which PCA is more suitable than LDA (d). Note we choose $a=1,2,3$ in this paper and plot the average solution.

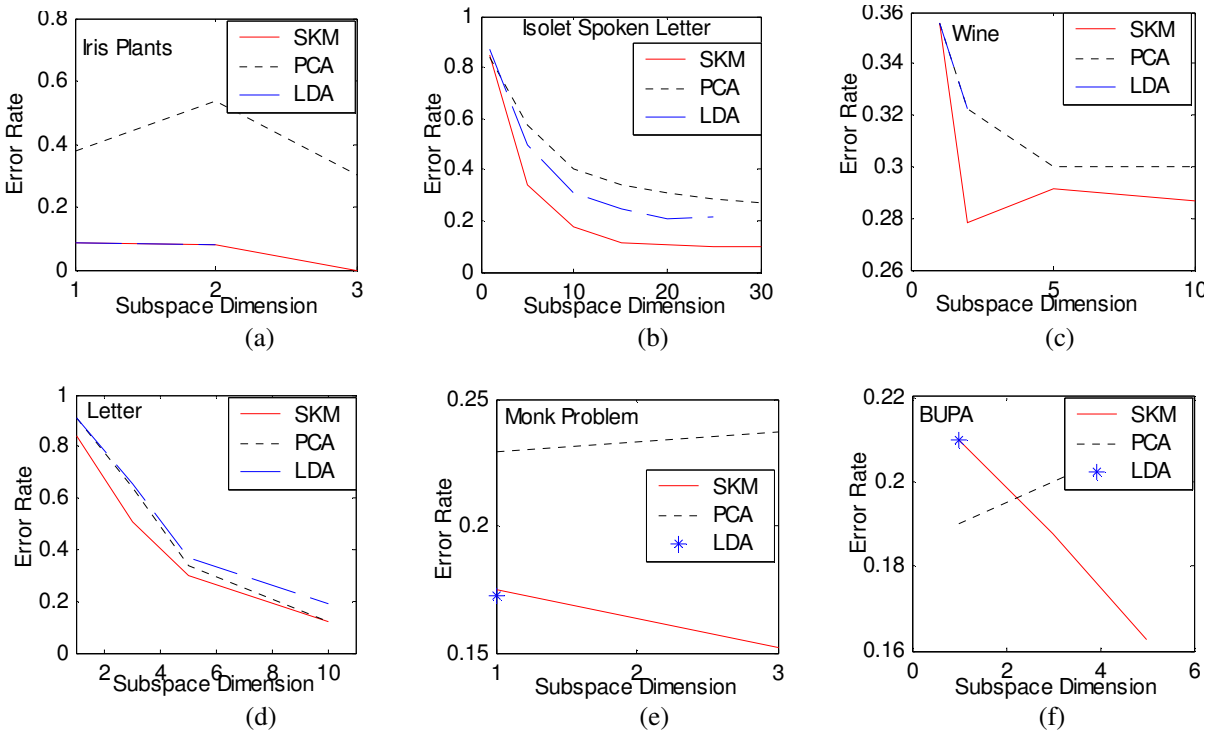

Figure 5. The error rate of some subsets of UCI by perform different subspace learning algorithms.

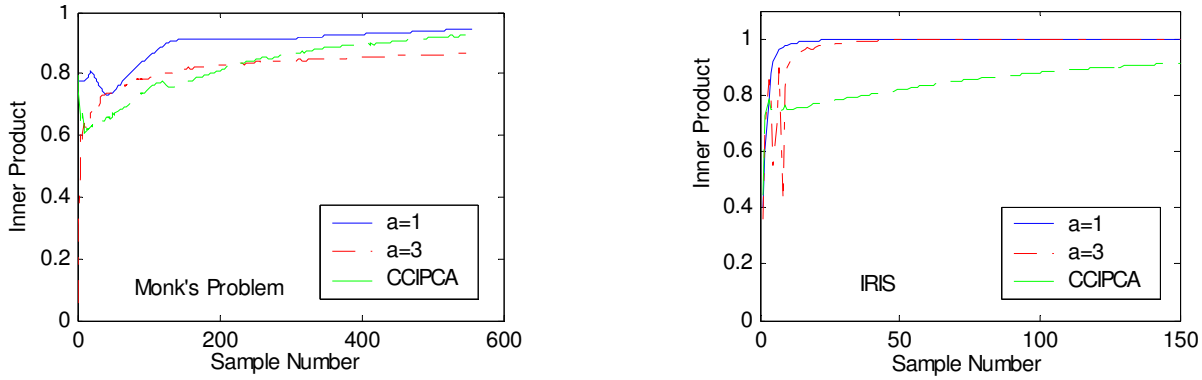

Figure 6. Convergence curve of incremental SKM

To measure the convergence ability of incremental SKM, we show the convergence curve on two subsets of UCI, IRIS and Monk's Problem, for intuition. Since $\|v-v'\| = 2(1-v\cdot v')$, and $v = v'$ iff. $v\cdot v' = 1$, the correlation between two unit eigen-vectors is represented by their inner product, and the larger the inner product is, the more similar the two eigen-vectors are. Figure 6 shows the inner product between eigen-vectors found by the proposed incremental algorithm in each step and the eigen-vectors found by the batch approach. The x-axis is the number of training data and the y-axis is the inner-product. From this Figure we can see that, incremental SKM can converge very fast. The CCIPCA [13] is the incremental PCA algorithm involved in this paper which has been used in section 5.1.

7. Conclusion and Future Work

In this paper, we proposed a novel supervised subspace learning algorithm called as Supervised Kampong Measurement. The incremental algorithm of this new criterion is also presented. In contrast to traditional LDA, the available subspace dimension with this measurement is not limited by the number of the classes and it is efficient and has no singularity problem in computation. Moreover, it is a highly scalable algorithm. In other words, it can process large scale data incrementally. The extensive

experiments on both synthetic and real datasets demonstrated that it outperforms PCA and even LDA on classification tasks. One of our future work is to give an algorithm which can learn the optimal parameter a automatically in the designing of weights.

8. References

[1]Al-Harbi, S.H. and Rayward-Sm, V.J. The Use of a Supervised k-Means Algorithm on Real-Valued Data with Applications in Health. *Lecture Notes in Computer Science*, *Volume 2718 / 2003*. 575 - 581.

[2]Artae, M., Jogan, M. and Leonardis, A., Incremental PCA for On-line Visual Learning and Recognition. In *Proceedings of the 16th International Conference on Pattern Recognition*, (Quebec City, QC, Canada, 2002), 781-784.

[3]Asai, T., Arimura, H., Abe, K., Kawasoe, S. and Arikawa, S., Online Algorithms for Mining Semi-structured Data Stream. In *Proceedings of the Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, (Maebashi City, Japan, 2002), 27.

[4]Balakrishnama, S. and Ganapathiraju, A. Linear Discriminant Analysis - A brief Tutorial, Institute for Signal and Information Processing, MS, 1998.

[5]C.L., B. and C.J., M. UCI Repository of machine learning databases Irvine. *CA: University of California, Department of Information and Computer Science.*

[6]Domingos, P. and Hulten, G., Mining High-Speed Data Streams. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Boston, MA, 2000), 71-80.

[7]La, F.D., Torre and J.Black, M. A Framework for Robust Subspace Learning. *International Journal of Computer Vision*, *54* (1-3).

[8]Lewis, D., Yang, Y., Rose, T. and Li, F. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*.

[9]Li, H., Jiang, T. and Zhang, K., Efficient and Robust Feature Extraction by Maximum Margin Criterion. In *Proceedings of the Advances in Neural Information Processing Systems 16*, (Vancouver, Canada, 2004), MIT Press.

[10]Martinez, A.M. and Kak, A.C. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23* (2). 228-233.

[11]Torkkola, K., Linear Discriminant Analysis in Document Classification. In *Proceedings of the*, (2001), 800-806.

[12]Wagstaff, K., Cardie, C., Rogers, S. and Schr, S., Constrained K-means Clustering with Background Knowledge. In *Proceedings of the ICML*, (Williams College, Williamstown, MA, USA, 2001), 577-584.

[13]Weng, J., Zhang, Y. and Hwang, W.-S. Candid Covariance-free Incremental Principal Component Analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *25* (8). 1034-1040.

[14]Wu, A., Mining Data Streams: A Survey of Algorithms and Applications. In *Proceedings of the http: //www-courses.cs.uiuc.edu/~cs412/slides/Stream1AWu.pdf*.

[15]Yan, J., Zhang, B.Y., Yan, S.C., Chen, Z., Fan, W.G., Yang, Q., Ma, W.Y. and Cheng, Q.S., IMMC: Incremental Maximum, Marginal Criterion. In *Proceedings of the To Appear in SIGKDD'04*, (Seattle,WA, 2004).

[16]Yu, H. and Yang, J. A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34. 2067-2070.

[17]Yang, Y. and Pedersen, J.O., A comparative Study on Feature Selection in Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, (1997), 412-420

Appendix

Theorem 1, Weights (13) satisfies all the constraints listed in section 4.2 when $a \geq 1$,

Proof: Since $a \geq 1$ and $0 \leq p_j \leq 1$, we can obviously seen from (13) that if $j \neq l_i$, $e_{l_i,j}$ increases when p_j increases; on the other hand, if $j = l_i$, $e_{l_i,j}$ decreases when p_j increases. Thus the first two constrains are satisfied.

The third constrain is a common one. $e_{l_i,j} \geq 0$ due to $a \geq 1$ and M is a positive number used to normalize the weights.

For the latest one, since $\sum p_j = 1$ and

$$e_{l_i,l_i} - \sum_{j \neq l_i} e_{l_i,j} = (a - p_{l_i}) - (1 - p_{l_i}) = a - 1$$

The latest constrain is satisfied if and only if $a \geq 1$.

End of Proof.