An Incremental Subspace Learning Algorithm to Categorize Large Scale Text Data

Jun Yan[1], Qiansheng Cheng[1], Qiang Yang[2], and Benyu Zhang[3]

[1] LMAM, Department of Information Science, School of Mathematical Sciences,
Peking University, Beijing, P.R. China 100871
yanjun@math.pku.edu.cn, qcheng@pku.edu.cn
[2] Department of Computer Science,
Hong Kong University of Science and Technology, Hong Kong
qyang@cs.ust.hk
[3] Microsoft Research Asia, 49 Zhichun Road, Beijing, P.R. China 100080
byzhang@microsoft.com

Abstract. The dramatic growth in the number and size of on-line information sources has fueled increasing research interest in the incremental subspace learning problem. In this paper, we propose an incremental supervised subspace learning algorithm, called Incremental Inter-class Scatter (IIS) algorithm. Unlike traditional batch learners, IIS learns from a stream of training data, not a set. IIS overcomes the inherent problem of some other incremental operations such as Incremental Principal Component Analysis (PCA) and Incremental Linear Discriminant Analysis (LDA). The experimental results on the synthetic datasets show that IIS performs as well as LDA and is more robust against noise. In addition, the experiments on the Reuters Corpus Volume 1 (RCV1) dataset show that IIS outperforms state-of-the-art Incremental Principal Component Analysis (IPCA) algorithm, a related algorithm, and Information Gain in efficiency and effectiveness respectively.

1 Introduction

In the last decades, the emergence of the daily growth of databases on the Web classification or the face recognition has revived the old problem of incremental and on-line algorithm of subspace learning [5, 14]. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two most popular linear subspace learning algorithms [2, 6, 10-12, 18].

PCA is an unsupervised subspace learning algorithm. It aims at finding out the geometrical structure of data set and projecting the data along the directions with maximal variances. However, it discards the class information which is significant for classification tasks. Through Singular Value Decomposition (SVD)[9], PCA can find an optimal subspace in the sense of least square reconstruction error. Its computational complexity is $O(m^3)$, where m is the minor value between the sample number and the data dimension. LDA is a supervised subspace learning algorithm. It searches for the projection axes on which the data points of different classes are far from each

Y. Zhang et al. (Eds.): APWeb 2005, LNCS 3399, pp. 52 – 63, 2005.

other and at the same time where the data points of the same class are close to each other. Unlike PCA which encodes information in an orthogonal linear space, LDA encodes discriminating information in a linear separable space whose bases are not necessarily orthogonal.

The original PCA is a batch algorithm, which means that the data must be given once altogether. However, this type of batch algorithms no longer satisfies the applications that the data are incrementally received from various data sources, such as online sensors [13]. Thus, an incremental method is highly desired to compute adaptive subspace for the data arriving sequentially. Incremental Principal Component Analysis (IPCA) [1, 16] are designed for such a purpose and have been studied for a long time. However, IPCA ignores the valuable class label information of the training data and the most representative features derived from IPCA may not be the most discriminant ones. The Incremental Support Vector Machine (ISVM) techniques have been developed fleetly. But most of them are approximate and require several passed through the data to reach convergence. Researchers [3, 4] have proposed incremental supervised learning based on neural network [4], but the algorithm convergence and stability still remain questionable.

In this paper, we propose an incremental supervised subspace learning algorithm based on statistical efficiency by *incrementally* optimizing the *Inter-class Scatter* criterion, so-call *IIS*. It derives the online adaptive supervised subspace using data samples received sequentially and incrementally updates the eigenvectors of the inter-class scatter matrix. IIS does not need to reconstruct the inter-class scatter matrix whenever it receives new sample data, thus it is very fast computationally. We also proved the convergence of the algorithm in this paper. The experimental results on the synthetic datasets show that IIS can learn a subspace similar to but more robust than LDA; and the experimental results on a real text dataset, Reuters Corpus Volume 1 (RCV1) [8], compared with IPCA and Information Gain (IG) demonstrate that IIS yields significantly better micro F1 and macro F1 than two baseline algorithms – IPCA and Information Gain (IG).

The rest of the paper is organized as follows. We present the incremental subspace learning algorithm IIS and the proof of convergence in section 2. Then, we demonstrate the experimental results on the synthetic datasets and the real word data, the Reuter Corpus Volume 1 in Section 3. We conclude our work in Section 4.

2 Incremental Supervised Subspace Learning

As Introduced above, IPCA ignores the class label information and the most representative features found by IPCA are not always the most discriminating features. This motivates us to design a supervised subspace learning algorithm that efficiently utilizes the label information. In this work, we consider the scenario to maximize the Inter-class scatter criterion that aims to make the class centers as far as possible.

Denote the projection matrix from original space to the low dimensional space as $W \in R^{d \times p}$. In this work, we propose to incrementally maximize the Inter-class scatter (IIS) criterion $J_s = W^T S_b W$, where $S_b = \sum_{i=1}^{c} p_i (m_i - m)(m_i - m)^T$ is the inter-class scatter matrix the same as in LDA. It is obvious that W is the first k leading eigenvectors of the matrix S_b and the column vectors of W are orthogonal to each other.

In the following subsections, we will present the details on how to incrementally derive the leading eigenvectors of S_b ; then the convergence proof and algorithm summary are also presented.

2.1 The First Eigenvector

Lemma-1: if $\lim_{n\to\infty} a_n = a$ then $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} a_i = a$.

Assume that a sample sequence is presented as $\{x_{l_n}(n)\}$, where *n=1, 2*.... The purpose of IIS is to maximize the Inter-class scatter criterion $J_s(W) = W^T S_b W$. Here k is the dimension of transformed data, i.e. the final subspace dimension.

The Inter-class scatter matrix of step n after learning from the first n samples can be written as below,

$$S_b(n) = \sum_{j=1}^{c} \frac{N_j(n)}{n}(m_j(n) - m(n))(m_j(n) - m(n))^T$$

From the fact that $\lim_{n\to\infty} S_b(n) = S_b$ and the lemma-1, we obtain

$$S_b = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} S_b(i) \tag{1}$$

The general eigenvector form is $Au = \lambda u$, where u is eigenvector corresponding to the eigenvalue λ . By replacing the matrix A with the Inter-class scatter matrix at step we can obtain an approximate iterative eigenvector computation formulation with $v = \lambda u$:

$$\begin{aligned} v(n) &= \frac{1}{n}\sum_{i=1}^{n} S_b(i)u(i) \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} \frac{N_j(i)}{i}(m_j(i) - m(i))(m_j(i) - m(i))^T u(i) \\ &= \frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{c} p_j(n)\Phi_j(i)\Phi_j(i)^T)u(i) \end{aligned}$$

where $\Phi_j(i) = m_j(i) - m(i)$, $v(n)$ is the n^{th} step estimation of v and $u(n)$ is the n^{th} step estimation of u . Once we obtain the estimate of v , eigenvector u can be directly computed as $u = v/\|v\|$. Let $u(i) = v(i-1)/\|v(i-1)\|$, we have the following incremental formulation:

$$v(n) = \frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{c} p_j(n)\Phi_j(i)\Phi_j(i)^T)v(i-1)\Big/\|v(i-1)\|$$

i.e. $$v(n)=\frac{n-1}{n}v(n-1)+\frac{1}{n}(\sum_{j=1}^{c}p_j(n)\Phi_j(n)\Phi_j(n)^T)v(n-1)\big/\|v(n-1)\|$$

the formula can be rewritten as:

$$\begin{aligned}v(n)&=\frac{n-1}{n}v(n-1)+\frac{1}{n}(\sum_{j=1}^{c}\Phi_j(n)\Phi_j(n)^T)v(i-1)\big/\|v(i-1)\|\\&=\frac{n-1}{n}v(n-1)+\frac{1}{n}\sum_{j=1}^{c}p_j(n)\alpha_j(n)\Phi_j(n)\end{aligned}$$

where $\alpha_j(n)=\Phi_j(n)^T v(i-1)\big/\|v(i-1)\|$, $j=1,2,...,c$.

For initialization, we set $v(0)$ as the first sample. Through this way, the subspace directions, i.e. the eigenvectors to be solved at time step *n* could be computed by the eigenvectors at time step *n-1* and the new arrived data at time step *n*.

2.2 Higher-Order Eigenvectors

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

$$x_{l_n}^{j+1}(n)=x_{l_n}^{j}(n)-(x_{l_n}^{j}(n)^T v^j(n))v^j(n)/\left\|v^j(n)\right\|^2$$

where $x_{l_n}^{1}(n)=x_{l_n}(n)$. The same method is used to update $m_i^j(n)$ and $m^j(n)$ $i=1,2,...,c$ .Since $m_i^j(n)$ and $m^j(n)$ are linear combination of $x_{l_i}^{j}(i)$, where $i=1,2,...,n$, $j=1,2,...,k$, and $l_i\in\{1,2,...,C\}$, Φ_i are linear combination of m_i and m , for convenience, we can only update Φ at each iteration step by

$$\Phi_{l_n}^{j+1}(n)=\Phi_{l_n}^{j}(n)-(\Phi_{l_n}^{j}(n)^T v^j(n))v^j(n)/\left\|v^j(n)\right\|^2$$

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.

Through the projection procedure at each step, we can get the eigenvectors of S_b one by one. It is much more efficient compared with the time-consuming orthonor malization process.

2.3 Convergence Proof

Lemma-2: Let $A(n)=S_b(n)$, $A=S_b$, then for any large enough N

$$\lim_{n\to\infty}p\{\sup\left\|\frac{A(n)-A}{\|v(n-1)\|}v(n-1)\right\|\geq\varepsilon\}=0$$

Lemma-3: $v(n)$ is bounded with probability 1.

Theorem: Let v^* be a locally asymptotically stable (in the sense of Liapunov) solution to the Ordinary Differential Equation bellow:

$$\overset{\bullet}{v} = (\frac{A}{\|v\|} - I)v$$

with domain of attraction $D(v^*)$. If there is a compact set $\varphi \in D(v^*)$ such that the solution of the equation (**) below satisfies $P\{v(n) \in \varphi\} = 1$, then $v(n)$ converges to v^* almost surely. **Note**:

$$\begin{aligned} v(n) \quad &= v(n-1) + \frac{1}{n}(A-I)v(n-1) + \frac{1}{n}(A(n)-A)v(n-1) \qquad ** \\ &= \frac{n-1}{n}v(n-1) + \frac{1}{n}u(n)u^T(n)v(n-1) \qquad if \qquad \|v(n-1)\| = 1 \end{aligned}$$

The convergence is a classical result from the theorems of stochastic approximation [7]. From the lemmas and theorem we can draw the conclusion of convergence [20].

Table 1. Algorithm Summary

```
for $n = 1,2,...$, do the following steps,
Update $N_i(n), m_i(n), \Phi_i(n), m(n)$ following the aforementioned steps;
  $\Phi_i^1(n) = \Phi_i(n) \qquad i = 1,2,...,c$
for $j = 1,2,..., \min\{K, n\}$
if j $= n$ then
    $v_j(n) = u_i^j(n)$ ,
else
    $\alpha_i^j(n) = \Phi_i^j(n)^T v^j(n-1) / \|v^j(n-1)\|^2$
      $v^j(n) = \frac{n-1}{n} v^j(n-1) + \frac{1}{n} \sum_{i=1}^{c} p_i^j(n) \alpha_i^j(n) \Phi_i^j(n)$
    $\Phi_i^{j+1}(n) = \Phi_i^j(n) - \Phi_i^j(n)^T v^j(n) v^j(n) / \|v^j(n)\|^2$
    $u_i^{j+1}(n) = u_i^j(n) - u_i^j(n)^T v^j(n) v^j(n) / \|v^j(n)\|^2$
End
```

2.4 Algorithm Summary

Suppose that at *Step* n , $x_{l_n}(n)$ is the input sample, which belongs to class l_n , $l_n \in \{1,2,...,c\}$,. $N_i(n)$ is the total sample number of class i . $m_i(n)$ is the mean of class i . $m(n)$ is the mean of all samples. K is the dimension of subspace to be found by our algorithm. Set $\Phi_j(i) = m_j(i) - m(i)$. The full algorithm is as table 1. The solution of step n is $v_j(n)$, $j = 1,2,...,K$.

2.5 Algorithm Property Analysis

The time complexity of IIS 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, IIS 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, IIS is able to handle large scale and continuous data.

IIS is also robust since IIS focuses on the mean of each class and all samples. That means that a little amount of mislabeled data could not affect the final solution. In fact, the robustness is determined by the criterion itself.

3 Experimental Results

We performed two sets of experiments. For intuition, in the first set of experiments, we used synthetic data that follow the normal distribution to illustrate the subspaces learned by IIS, LDA, and PCA, along with performance in a noise data. Since the web documents are large scale text data, thus to demonstrate the performance of our proposed algorithm on large scale text data, in the second set of experiments, we applied several dimension reduction methods on the Reuters Corpus Volume 1 (RCV1) dataset, and then compare the classification performance and the time cost. Reuters Corpus Volume 1 data set [8] contains over 800,000 documents. Moreover, each document is represented by a vector with the dimension about 300,000.

3.1 Synthetic Data

We generated a 2-dimension data set of 2 classes. Each class consists of 50 samples by following normal distribution with means (0, 1) and (0,-2), respectively; and the covariance matrix of them are *diag*(1, 25) and *diag*(2, 25). Figure 1 shows a scatter plot of the data set, along with the 1-d subspace learned by IIS, LDA, and PCA,

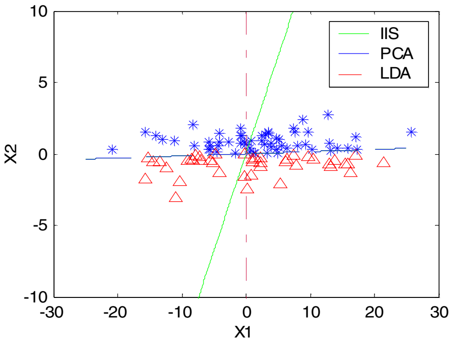

Fig. 1. Subspaces learned by IIS, LDA, and PCA

represented by the straight line, the dash dotted line, and the broken line, respectively. We can see that IIS can outperform PCA and yield comparable performance to LDA for classification.

To demonstrate the robustness of IIS against noise, we generated a 3-d data set of 3 classes each of which consists of 200 samples that follows the normal distribution with means (0, 5, 5), (0, 5, 10), and (5, 5, 10) and the same covariance matrixes Diag(5, 5, 5), where we performed IIS and LDA to learn the 2-d eigenspaces. We randomly provided several abnormal samples, and then compared the correlation between the "noise" eigenvectors and the "original" eigenvectors of each algorithm. Since $\|v - v'\| = 2(1 - v \cdot v')$, and $v = v'$ iff $v \cdot v' = 1$, the correlation between two unit eigenvectors is represented by their inner product, and the larger the inner product is, the more robustness against noise. The results are shown in Table 2.

Table 2. Correlation between the "noise" eigenvectors and the "original" eigenvector learned by IIS and LDA

Mislabeled Data Per Class	5	10	15	20
1ST Eigenvector of IIS	1	0.9999	0.9970	0.9963
2RD Eigenvector of IIS	1	0.9999	0.9990	0.9960
1ST Eigenvector of LDA	0.9577	0.8043	0.7073	0.6296
2RD Eigenvector of LDA	1	0.9992	0.9968	0.9958

We can see from table 1 that IIS is more robust against noises than LDA. With 20 mislabeled data (=10%) for each class, IIS can keep the inner product bigger than 99.6%. The intuitive reason for LDA being sensitive to noise comes from that LDA processes the matrix $S_w^{-1}S_b$. A small amount of mislabeled data can make S_w change, and even very little change of S_w makes S_w^{-1} change a lot. In other words, $S_w^{-1}S_b$ is very sensitive to the change of samples' label, and therefore the eigenvectors of $S_w^{-1}S_b$ are very sensitive to abnormal data.

Though the IIS has good performance on the synthetic data, our motivation to design it is to reduce the dimension of very large scale web documents or other large scale data sets, we conduct it on the widely used large scale text data RCV1 to introduce IIS.

3.2 Real World Data

To compare the effectiveness and efficiency of IIS to that of other subspace learning algorithms, we constructed classification experiments on the Reuters Corpus Volume 1 (RCV1) data set [8] which contains over 800,000 documents. We choose 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 split them into 5 equal-sized subsets, and each time 4 of them are used as the training set and the remaining ones are left as the test set. The experimental results reported in this paper are the average of the five runs. In these experiments, we use a single computer with Pentium(R) 4 CPU 2.80GHz, 1GB of RAM, Microsoft Windows XP Professional Version, to conduct the experiments. The coding language used by us is C++ 7.0. The most widely used performance measurement for text categorization problems are Precision, Recall and F1. Precision is a proportion which could be computed by the number of right categorized data over the number of all testing data. Recall is a proportion which could be computed by the number of right categorized data over the number of all the assigned data. F1 is a common measure in text categorization that combines recall and precision. We use two different *F1* measurements, i.e. micro *F1* and macro *F1* in our paper.

3.2.1 Experiment Setup

The dimensionality reduction algorithms are applies in the following manner:

- Apply the dimensionality reduction algorithm on a specific size of the training data to learn a subspace;
- Transform all the training data to the subspace;
- Train SVM by SMO [15];
- Transforming all the test data to the subspace;
- Evaluate the classification performance, using F1 value, on the transformed test data.

The dimension reduction algorithms applied are:

- The proposed IIS generating a 3-d subspace. We applied IIS on the first 10, 100, 1,000, 10,000, and 100,000 training data to study the convergence speed.
- Information Gain (IG). This is a state-of-the-art text classification method [17]. In this paper, we applied IG on all training data to generate 3-d and 500-d subspaces, denoted by IG3 and IG500, respectively. With the same dimension, IG3 performs as effective as ISBC; while IG500 will yields almost best classification performance, since SVM is insensitive to the number of feature [19].
- IPCA following the CCIPCA algorithm [16]. We also used IPCA to generate both 3-d and 500-d subspaces.

3.2.2 Effectiveness of IIS

The classification performances are summarized in Figure *2* and Figure *3*. From these figures, we can infer that the eigenspace learned by IIS on 100 input samples is significantly better than the ones learned by IPCA and IG3; and after learning 100,000 input samples (<20%), IIS can generate a comparable eigenspace to the one generated by IG500 in terms of classification performance. Hence, IIS is an effective subspace learning algorithm for classification tasks. On the other hand, we can see that IIS generated a near optimal eigenspace after just learning 10,000 samples. This indicates that in practice, the convergence speed of IIS is very fast.

The F1 value of each class is shown in Table 3. The inferior classification performance of ECAT is probably due to the uneven class distribution, as shown in Table 3.

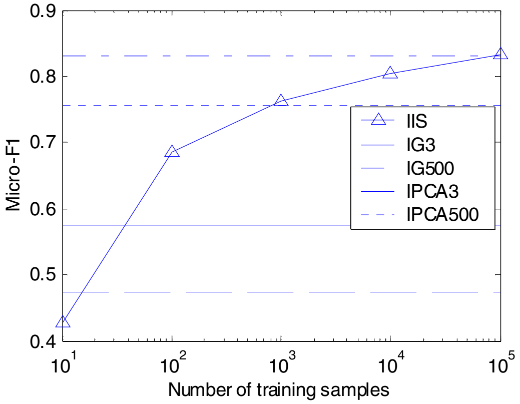

Fig. 2. Micro-F1 after reducing dimension by several subspace learning algorithms

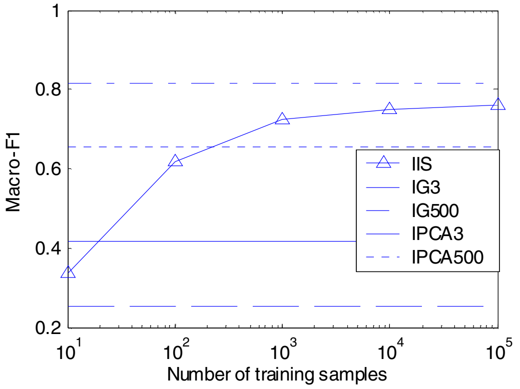

Fig. 3. Macro-F1 after reducing dimension by several subspace learning algorithms

3.2.3 Efficiency of IIS

The time spent of each algorithm in dimension reduction and classification training is reported in Table 4. We can see that the dimension reduction time of IIS is almost linear related to the number of input samples. Although the IG is faster than IIS, Its classification training time is much longer than that of IIS. The reason for IG500 being near square time complexity of SVM; while the possible reason for IG3 is that the optimization process of SVM is very slow if the margin between different classes is small which is unfortunately the case in the eigen-space of IG3.

Table 3. F1 values of each class using different dimension reduction algorithms

	CCAT	ECAT	GCAT	MCAT
IIS 3@10	0.596	0.117	0.443	0.192
IIS 3@100	0.711	0.213	0.762	0.793
IIS 3@1000	0.744	0.430	0.877	0.759
IIS 3@10000	0.873	0.443	0.875	0.850
IIS 3@100000	0.846	0.491	0.881	0.882
IG 3@ALL	0.696	0	0.524	0.451
IG 500@ALL	0.835	0.716	0.843	0.869
IPCA 3@ALL	0.632	0	0.376	0
IPCA 500@ALL	0.782	0.180	0.858	0.802
# SAMPLES (*10^5)	3.74	1.18	2.35	2.00

Table 4. Time costs (in seconds) of each dimension reduction algorithms

	DIMENSION REDUCTION TIME	CLASSIFICATION TRAINING TIME
IIS 3@10	1.85	1,298
IIS 3@100	17.1	11,061
IIS 3@1000	177	6,474
IIS 3@10000	2,288	7,560
IIS 3@100000	26,884	3,343
IG 3@ALL	136	52,605
IG 500@ALL	137	312,887
IPCA 3@ALL	28,960	25,374
IPCA 500@ALL	3,763,296	9,327

4 Conclusion and Future Works

In this paper, we proposed an incremental supervised subspace learning algorithm, IIS, which is a challenging issue of computing dominating eigenvectors and eigenvalues from incrementally arriving sample stream without storing the knowing data in advance. This proposed IIS algorithm is fast in convergence rate, low in the computational complexity, efficient, effective and robust. Experimental results on synthetic dataset and real text dataset demonstrate that it outperforms IPCA on classification tasks. It can be theoretically proved that IIS can find out the same subspace as LDA does if every class is uniformly distributed in all directions. In real word applications, this assumption can not always be satisfied; therefore intra-class scatter matrix in LDA is also very important for classification tasks. In the future work, we plan to extend the incremental supervised learning to consider both the inter-class and intra-class scatter matrices and we are currently exploring these extensions in theory and practice.

Acknowledgement

The authors would like to thank Ning Liu for the improvement of this paper.

References

[1] 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.

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

[3] Chatterjee, C. and Roychowdhury, V.P. On self-organizing algorithms and networks for class-separability features. *IEEE Trans. on Neural Networks*, *8* (3). 663 - 678.

[4] Hiraoka, K., Hidai, K., Hamahira, M., Mizoguchi, H., Mishima, T. and Yoshizawa, S., Successive Learning of Linear Discriminant Analysis: Sanger-Type Algorithm. In *Proceedings of the 14 th International Conference on Pattern Recognition*, (Barcelona, Spain, 2000), 2664-2667.

[5] Hoch, R., Using IR techniques for text classification in document analysis. In *Proceedings of the Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, (Dublin, Ireland, 1994), Springer-Verlag New York, Inc., 31 - 40.

[6] Jolliffe, I.T. *Principal Component Analysis*. Springer-Verlag, 1986.

[7] Kushner, H.J. and Clark, D.S. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York, 1978.

[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] M.W., B. Large-scale sparse singular value computations. *International Journal of Supercomputer Applications*, *6*. 13-49.

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

[11] Moghaddam, B. and Pentland, A. Probabilistic Visual Learning for Object Representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *19* (7). 696-710.

[12] Murase, H. and Nayar, S.K. Visual Learning and Recognition of 3D objects from Appearance. *International Journal of Computer Vision*, *14* (1). 5-24.

[13] Muthukrishnan, S. Data stream algorithms and applications. *Rutgers/AT&T Shannon Labs*.

[14] Oja., E. Subspace methods of pattern recognition. *Pattern recognition and image processing series.*, *6* (John Wiley & Sons).

[15] Platt, J. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: support vector learning*, MIT Press, Cambridge, MA, 1999, 185-208.

[16] 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.

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

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

[19] Zhang, J., Jin, R., Yang, Y. and Hauptmann, A.G., Modified Logistic Regression: An Approximation to SVM and Its Applications in Large-Scale Text Categorization. In *Proceedings of the 20 th International Conference on Machine Learning*, (Washington, DC, 2003), 888-895.

[20] Zhang, Y. and Weng, J. Convergence analysis of complementary candid incremental principal component Analysis, Michigan State University, 2001.