Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing

Jun Yan, Benyu Zhang, Ning Liu, Shuicheng Yan, Qiansheng Cheng, Weiguo Fan, Qiang Yang, Wensi Xi, and Zheng Chen

Abstract—Dimensionality reduction is an essential data preprocessing technique for large-scale and streaming data classification tasks. It can be used to improve both the efficiency and the effectiveness of classifiers. Traditional dimensionality reduction approaches fall into two categories: *Feature Extraction* and *Feature Selection*. Techniques in the feature extraction category are typically more effective than those in feature selection category. However, they may break down when processing large-scale data sets or data streams due to their high computational complexities. Similarly, the solutions provided by the feature selection approaches are mostly solved by greedy strategies and, hence, are not ensured to be optimal according to optimized criteria. In this paper, we give an overview of the popularly used feature extraction and selection algorithms under a unified framework. Moreover, we propose two novel dimensionality reduction algorithms based on the *Orthogonal Centroid* algorithm (OC). The first is an Incremental OC (IOC) algorithm for feature extraction. The second algorithm is an Orthogonal Centroid Feature Selection (OCFS) method which can provide optimal solutions according to the OC criterion. Both are designed under the same optimization criterion. Experiments on Reuters Corpus Volume-1 data set and some public large-scale text data sets indicate that the two algorithms are favorable in terms of their effectiveness and efficiency when compared with other state-of-the-art algorithms.

Index Terms—Feature extraction, feature selection, orthogonal centroid algorithm.

1 Introduction

Dimensionality reduction is an essential task for many large-scale information processing problems such as classifying document sets, searching over Web data sets, etc [20], [25], [29]. Due to the rapid growth of the World Wide Web, many traditional classification techniques require a huge amount of memory and CPU resource if dimensionality reduction were not performed well. For example, according to [25], a typical document classification system consists of tasks such as documents collection, vector space transformation [28], dimensionality reduction, classifier design, and system evaluation. Among all the above-mentioned components, dimensionality reduction is of great importance for the quality and efficiency of a classifier especially for large-scale real-time data since the bottleneck of the classification task is the poor classification efficiency caused by the high dimension of the feature space.

- *J. Yan, S. Yan, and Q. Cheng are with LMAM, Department of Information Science, School of Mathematical Science, Peking University, Beijing 100871, P.R. China. E-mail: {yanjun, scyan, qcheng}@math.pku.edu.cn.*
- *B. Zhang and Z. Chen are with Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, P.R. China. E-mail: {byzhang, zhengc}@microsoft.com.*
- *N. Liu is with the Department of Mathematics, Tsinghua University, Beijing, 100084, P.R. China. E-mail: liun01@mails.tsinghua.edu.cn.*
- *W. Fan and W. Xi are with the Virginia Polytechnic Institute and State University, 1220 University City Blvd., Blacksburg, VA 24060. E-mail: {wfan, xwensi}@vt.edu.*
- *Q. Yang is with the Department of Computer Science, Room 3562 (Lift 25/26), Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong. E-mail: qyang@cs.ust.hk.*

Manuscript received 24 Nov. 2004; revised 13 May 2005; accepted 27 July 2005; published online 18 Jan. 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDESI-0486-1104.

The traditional and the state-of-the-art dimensionality reduction methods can be generally classified into *Feature Extraction* (FE) [17], [18], [22] and *Feature Selection* (FS) [2], [5], [14], [36] approaches. In general, FE approaches are more effective than the FS techniques [26], [32], [35] (except for some particular cases) and they have shown to be very effective for real-world dimensionality reduction problems [6], [9], [17], [18]. These algorithms aim to extract features by projecting the original high-dimensional data into a lower-dimensional space through algebraic transformations. The classical FE algorithms are generally classified into linear and nonlinear algorithms. Linear algorithms [4], [13], such as Principal Component Analysis (PCA) [12], Linear Discriminant Analysis (LDA) [21], [32], and Maximum Margin Criterion (MMC) [18], aim to project the high-dimensional data to a lower-dimensional space by linear transformations according to some criteria. On the other hand, nonlinear algorithms [3], such as Locally Linear Embedding (LLE) [27], ISOMAP [30], and Laplacian Eigenmaps aim to project the original data by nonlinear transformations while preserving certain local information according to some criteria. In contrast to the nonlinear algorithms, linear ones are of more interest and in wider usage due to their efficiency. Thus, we focus on the linear approaches in this paper. However, both the linear and the nonlinear FE approaches suffer from the high computational complexity of online computational problems associated with streaming data, or large-scale data that are present nowadays. Thus, it is highly necessary to develop scalable incremental feature extraction algorithms.

Many scalable online FE algorithms have been proposed recently. Incremental PCA (IPCA) [1], [19] is a well-studied incremental learning algorithm. Many types of IPCA algorithms have been proposed. The main difference among them is the incremental representation of the covariance matrix. The latest version of IPCA is called Candid Covariance-free Incremental Principal Component Analysis (CCIPCA) [33]. However, IPCA ignores the valuable label information of data and is not optimal for general classification tasks. The Incremental Linear Discriminant Analysis (ILDA) [8] algorithm has also been proposed recently. However, the singularity problem of LDA and the instability of ILDA algorithm limit their applications. Another online FE algorithm called Incremental Maximum Margin Criterion (IMMC) [34] is proposed recently as well. However, the parameter estimation of IMMC is still an open issue.

In contrast to the FE algorithms, FS algorithms [5], [14], [36] have been widely used on large-scale data and online applications. This is due to the fact that the FS approaches are much more efficient than the traditional FE approaches. They aim at finding out a subset of the most representative features according to some criteria, i.e., the features are ranked according to their individual predictive power. To rank the features, there are many widely used approaches, such as *Information Gain* (IG), χ^2-test (CHI), and *Mutual Information* (MI) [36]. The feature selection problem can be seen as a search in a hypothesis space (set of possible solutions). Thus, the resulting algorithms are mostly greedy and it is very hard to find the global optimal solution by these algorithms. Finding the optimal feature selection solution is still a challenging issue, which we will solve in this paper.

From another perspective, dimensionality reduction approaches can be classified into supervised algorithms and unsupervised algorithms. Supervised approaches, such as LDA, MMC, and the Orthogonal Centroid algorithm (OC) [11], [23], need a training set with the class label information to learn the lower-dimensional representation according to some criteria, and then predict the class labels on unknown test data. The unsupervised approaches, such as PCA, project the original data to a new lower-dimensional space according to some criteria without utilizing the label information. Supervised approaches are usually more effective than unsupervised ones in classification capability, when the labeling information is available [21].

In this paper, we present a unified framework for both the FE and the FS algorithms, which give rise to two novel supervised dimensionality reduction algorithms. Both of them are under the same optimization criterion that is based on the Orthogonal Centroid algorithm (OC) [23], a recently proposed supervised FE approach through *QR* matrix decomposition [7]. The first one is an online FE algorithm which is called the Incremental OC (IOC) algorithm. It aims at finding out the optimal lower-dimensional representation of data adaptively according to the objective function implied by the Orthogonal Centroid criterion. The other one is a FS approach according to the same objective function, which we call Orthogonal Centroid Feature Selection (OCFS). The IOC algorithm reduces the complexity of classical batch FE approaches greatly and is simpler than both ILDA and IMMC for problems of large-scale data streams. In contrast to other FS algorithms, the OCFS can find the optimal solution according to the objective function given by Orthogonal Centroid algorithm, in a discrete solution space. Experimental results show comparable performance and complexity of our two novel algorithms with other competitive algorithms on Reuters Corpus Volume 1 (RCV1) data set [16] and other real large-scale data sets.

The rest of this paper is organized as follows: In Section 2, we introduce the background information and formulate our problem by introducing a unified framework of some popularly used dimensionality reduction techniques. In Section 3, we present two novel dimensionality reduction approaches and provide the theoretical analysis on them. In Section 4, the experimental results are given. Finally, we conclude this paper in Section 5.

2 Backgrounds and Problem Formulation

We introduce the background knowledge about dimensionality reduction in this section. Some notations and a unified optimization framework for supervised dimensionality reduction problem are given first. In Sections 2.2 and 2.3, we introduce some traditional Feature Extraction (FE) and Feature Selection (FS) approaches, respectively, under the same optimization model. In Section 2.4, we show the relationship between FE and FS intuitively.

2.1 Notations and the Dimensionality Reduction Problem

In this paper, a corpus of samples is mathematically represented by a $d \times n$ matrix $X \in R^{d \times n}$, where n is the number of objects and d is the feature number. Each object is denoted by a column vector $x_i, i = 1, 2, \cdots, n$, and the kth entry of x_i is denoted by $x_{ik}, k = 1, 2, \cdots, d$. X^T is used to denote the transpose of matrix X. Assume that these feature vectors belong to c different classes and the sample number of the jth class is n_j. We use c_j to represent class j, $j = 1, 2, \cdots, c$. The mean vector of the jth class is $m_j = \frac{1}{n_j} \sum_{x_i \in c_j} x_i$. The mean vector of all the samples is $m = \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{1}{n} \sum_{j=1}^{c} n_j m_j$. The dimensionality reduction problem can be stated as the problem of finding a function $f : R^d \rightarrow R^p$, where p is the dimension of data after dimensionality reduction ($p << d$), so that an object $x_i \in R^d$ is transformed into $y_i = f(x_i) \in R^p$. We consider the linear approaches only in this paper; thus, f should be a linear function. We formulate our framework of dimensionality reduction problem as follows:

Given a set of labeled training data X, learn a transformation matrix W such that W is optimal according to some objective function $J(W)$ in some given solution space.

Then, we can transform the unlabeled d-dimensional data by applying $y_i = W^T x_i$ and classify this unlabeled data in the p-dimensional space. We can show that the linear FE and FS approaches mentioned here can be formulated under

this framework. The differences between FE and FS are the different definitions of "solution space," as well as how objective functions are defined.

2.2 Feature Extraction

FE algorithms aim to extract features by projecting the original high-dimensional data to a lower-dimensional space through algebraic transformations. The classical FE algorithms are generally classified into linear and nonlinear approaches as mentioned in Section 1. In this paper, we focus on the linear approaches due to their efficiency for processing large-scale streaming data. From the FE perspective, following our defined dimensionality reduction framework, the problem is to find an optimal transformation matrix $W \in R^{d\times p}$ according to some criteria $J(W)$ such that $y_i = f(x_i) = W^T x_i \in R^p$, $i = 1, 2, \cdots, n$ are the p-dimensional representation of original data. We exercise freedom to multiply W with some nonzero constant. Thus, we additionally require that W consists of unit vectors. Then, the solution space is continuous and consisted of all the real $d \times p$ matrices subject to the constraint that $W^T W = I$, where I is an identity matrix. Note we use w to denote the column vector of W below.

The major differences among different FE algorithms are the different objective functions to learn the projection matrix $W \in R^{d\times p}$. We can show that the four popular linear FE algorithms, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Maximum Margin Criterion (MMC), and the Orthogonal Centroid (OC) algorithm are all under the same framework.

2.2.1 Principal Component Analysis

The goal of PCA is to find a subspace whose basis vectors correspond to the directions with maximal variances. Let's denote $C = \frac{1}{n}\sum_{i=1,2,\cdots,n}(x_i - m)(x_i - m)^T$ as the covariance matrix of sample data. We define the objective function as $J(W) = traceW^T CW$. Then, PCA aims to maximize the objective function $J(W)$ in a solution space $H^{d\times p} = \{W \in R^{d\times p}, W^T W = I\}$. It can be proven that the column vectors of W are the p leading eigenvectors of the covariance matrix C. The computation cost of PCA mainly lies in the Singular Value Decomposition (SVD) [31] processing with time complexity of $O(t^3)$, where $t = \min\{d, n\}$.

For large-scale data and data streams, Incremental PCA (IPCA) [1], [33] is a well-studied incremental learning algorithm. Many types of IPCA have been proposed. The main difference is the incremental representation of the covariance matrix. Though PCA can find the most representative features, it ignores the valuable class label information and, thus, is not optimal for general classification tasks.

2.2.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is used to find a lower-dimensional space that best discriminates the samples from different classes. It aims to maximize the Fisher criterion, i.e., an objective function:

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

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

$$S_w = \sum_{i=1}^{c} p_i \underset{x\in c_i}{E}\{(x - m_i)(x - m_i)^T\}$$

are called Interclass scatter matrix and Intraclass scatter matrix, respectively. The E denotes the expectation and $p_i = n_i/n$ is the prior probability for a sample belonging to class i. W can be obtained by solving $W^* = \arg\max J(W)$ in solution space $H^{d\times p} = \{W \in R^{d\times p}, W^T W = I\}$. This can be done by solving the following generalized eigenvalue decomposition problem: $S_b w = \lambda S_w w$.

LDA explicitly utilizes the label information of the samples which is suitable for classification problems. However, since there are at most $c - 1$ nonzero eigenvalues, the upper bound of p is therefore $c - 1$; and at least $d + c$ sample data is required to make it possible that S_w is not singular, which limits the application of LDA. Moreover, it is still difficult for LDA to handle databases with high-dimensional representation or streaming data. As in the Reuters Corpus Volume 1 [16], the data dimension is about 300,000 and it is hard to conduct SVD efficiently. The Incremental Linear Discriminant Analysis (ILDA) [8] is designed to solve this problem. However, the stability of ILDA is still an issue as in our experiments.

2.2.3 Maximum Margin Criterion

Maximum Margin Criterion (MMC) [18] is a recently proposed supervised FE algorithm. Based on the same representation as LDA, MMC aims to maximize the objective function:

$$W^* = \arg\max J(W) = \arg\max\ trace\{W^T(S_b - S_w)W\},$$
$$W \in H^{d\times p}.$$

Although both MMC and LDA are supervised subspace learning approaches, the computation of MMC is easier than that of LDA since MMC does not have inverse matrix operation. The projection matrix W can be obtained by solving the eigenvalue decomposition problem: $(S_b - S_w)w = \lambda w$.

Similar to other batch feature extraction approaches, MMC is not efficient for large-scale data or streaming data problems. An Incremental Maximum Margin Criterion (IMMC) [34] algorithm was proposed recently in response to this problem. However, this method is sensitive to parameter settings, which is an open problem.

2.2.4 Orthogonal Centroid Algorithm

Orthogonal Centroid (OC) algorithm [11], [23] is a recently proposed supervised FE algorithm which utilizes orthogonal transformation on centroid. It has been proven to be very effective for classification problems [10] and is based on the vector space computation in linear algebra [7] by *QR* matrix decomposition. The Orthogonal Centroid algorithm for dimensionality reduction has been successfully applied on text data [11]. However, the time and space cost of *QR* decomposition are too expensive for large-scale data such as Web documents. To address this issue, in this

paper, we propose not only an incremental Orthogonal Centroid algorithm, but also a FS algorithm which can give the optimal solution based on OC. Theorem 1 below shows that the OC algorithm can be derived from the framework defined in the previous section.

Theorem 1. *The solution of Orthogonal Centroid algorithm equals to the solution of the following optimization problem,*

$$W^* = \arg\max J(W) = \arg\max\ trace(W^T S_b W),$$
$$W \in H^{d\times p} = \{W \in R^{d\times p}, W^T W = I\}.$$

The detailed proof of this theorem can be found in [10], [23]. This objective function aims at separating different classes as far as possible in the transformed lower-dimensional space.

2.3 Feature Selection

In contrast to the FE algorithms, FS algorithms [5], [14], [36] have been widely used on large-scale data and online learning problems due to their efficiency. They aim at finding out a subset of the most representative features according to some objective function. According to [36], Information Gain (IG) and CHI are two of the most classical and effective feature selection algorithms. Thus, we involve them as baselines in this paper.

From the FS point of view, the purpose of dimensionality reduction is to find a subset of features indexed by $k_l, l = 1, 2, \cdots, p$ such that the lower-dimensional representation of original data x_i is denoted by $y_i = f(x_i) = (x_{ik_1}, x_{ik_2}, \cdots, x_{ik_p})^T$. Note that each feature index set corresponds to a unique binary matrix, i.e., the $y_i = (x_{ik_1}, x_{ik_2}, \cdots, x_{ik_p})^T$ can be achieved by $y_i = \tilde{W}^T x_i$, where $\tilde{W}$ is a 0-1 binary matrix with $w_{k_i i} = 1$, $i = 1, 2, \cdots, p$ and others equal to zero. Then, following the same framework with the previously introduced FE algorithms, the FS problem is to find an optimal transformation matrix $\tilde{W} \in R^{d\times p}$ according to some criteria $J(\tilde{W})$ subject to the constraint that $\tilde{W} = \{\tilde{w}_{ik}\}$ is a binary matrix whose entries are equal to zero or one and each column of $\tilde{W}$ has a unique nonzero element. Then, the low-dimensional representation of original data is $y_i = \tilde{W}^T x_i = (x_{ik_1}, x_{ik_2}, \cdots, x_{ik_p})^T$. The solution space of the feature selection problem is discrete and can be defined as

$$\tilde{H}^{d\times p} = \{\tilde{W} = \{w_{ik}\} \in R^{d\times p},$$
$$w_{ik} \in \{0, 1\} \text{ for all } i \text{ and } k,$$
$$\text{if } w_{ik_j} = 1, \text{then } w_{ik_t} = 0 \text{ for all } t \neq j\}.$$

2.3.1 Information Gain

Information gain of a selected group of features $T = (t_{k_1}, t_{k_2}, \cdots, t_{k_p})$ could be calculated by:

$$IG(T) = -\sum_{j=1}^{c} P_r(c_j)\log P_r(c_j) + P_r(T)\sum_{j=1}^{c} P_r(c_j|T)\log P_r(c_j|T) + P_r(T)\sum_{j=1}^{c} P_r(\tilde{c_j}|T)\log P_r(\tilde{c_j}|T),$$

where t_{k_i} is used to denote a unique feature, $IG(T)$ is the information gain of a feature group, $P_r(c_j)$ is the probability of class c_j, $P_r(T)$ is the probability of feature group T and $P_r(c_i|T)$ is the corresponding conditional probability. Following the problem definition in Section 2.1, we define $J(\tilde{W}) = IG(T)$. In other words, IG aims to find an optimal $\tilde{W} \in \tilde{H}^{d\times p}$ according to $J(\tilde{W}) = IG(T)$ so that each object is represented by p features after the projection $y_i = \tilde{W}^T x_i$, then these p features could maximize $J(\tilde{W}) = IG(T)$. However, in practice, this is an NP problem and a greedy approach is typically used.

Given training objects, we compute the information gain of each feature t_{k_i} by:

$$IG(t_{k_i}) = -\sum_{j=1}^{c} P_r(c_j)\log P_r(c_j) + P_r(t_{k_i})\sum_{j=1}^{c} P_r(c_j|t_{k_i})\log P_r(c_j|t) + P_r(t)\sum_{j=1}^{c} P_r(\tilde{c_j}|t_{k_i})\log P_r(\tilde{c_j}|t_{k_i}).$$

Then, we remove those features whose information gain is less than some predetermined threshold. Obviously, the greedy IG is not optimal according to $J(\tilde{W}) = IG(T)$. The complexity of the greedy IG is $O(cd)$, where c is the number of classes.

2.3.2 CHI

CHI also aims at maximizing a criterion $J(\tilde{W}) = \chi^2(T)$, where $T = (t_{k_1}, t_{k_2}, \cdots, t_{k_p})$ is a selected group of features. To save the computation cost, CHI shares the same idea with the introduced IG. Instead of considering a group of features together, to a given feature t and a category c_j, suppose A is the number of times t and c_j co-occur, B is the number of times the t occurs without c_j, C is the number of times c_j occurs without t, and D is the number of times neither c_j nor t occurs. The χ^2 statistics is:

$$\chi^2(t, c_j) = \frac{n(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)}.$$

We can compute the χ^2 statistics between each unique feature and each category in a training corpus, and then combine the category specific scores of each feature into $\chi^2(t) = \sum_{j=1}^{c} P_r(c_j)\chi^2(t, c_j)$. Then, we remove those features whose χ^2 statistics are less than some predetermined threshold. It is also a greedy algorithm and, thus, not always optimal either. The computational complexity of IG and CHI are very similar. The main computational time is spent on the evaluation of the conditional probability and $\chi^2(t, c_j)$, respectively.

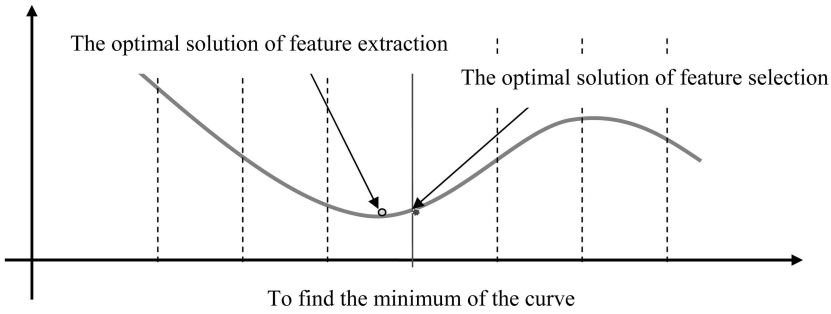


Fig. 1. The optimal solution of feature selection and feature extraction.

2.4 Feature Extraction versus Feature Selection

In this section, we discuss the relationship between FE and FS approaches. As shown above, all the linear FE and FS approaches involved can be formulated in a unified optimization framework. Different detailed FE or FS algorithms are derived from by different forms of objective functions. For a given objective function $J(W)$, FE algorithms aim at optimizing it in a continuous solution space $H^{d\times p} = \{W \in R^{d\times p}, W^T W = I\}$ and the FS algorithms aim at optimizing the objective in a binary discrete solution space

$$\begin{aligned}\tilde{H}^{d\times p} =& \{\tilde{W} = \{w_{ik}\} \in R^{d\times p},\\ & w_{ik} \in \{0,1\} \text{ for all } i \text{ and } k,\\ & \text{if } w_{ik_j} = 1, w_{ik_t} = 0 \text{ for all } t \neq j\}.\end{aligned}$$

Since both FE and FS are optimization problems, Fig. 1 helps learn the relationship between them. Suppose we want to use both FE and FS to minimize the same objective function which is denoted by the curve in Fig. 1, the minimal point in the continuous space (the space in which FE algorithms find their solution) is clearly given. However, this minimal point may not be reached in the discrete space (the space in which feature selection algorithms find their solution) which is described by the vertical lines. In the discrete space, the optimal solution is the intersecting point of the solid vertical line and the curve.

In other words, FE by linear algebraic transformation can find the optimal solution of a problem, which is not always true for FS algorithms. Moreover, the algorithms of FS are always greedy. Thus, they sometimes cannot even find the "optimal solution" in the discrete space. However, the computational complexity of FS techniques is always much lower than that of FE algorithms since there is no need to perform algebraic transformations. Due to its computational attractiveness, FS approaches are more popular than the FE techniques on large-scale data or streaming data for dimensionality reduction.

3 New Dimensionality Reduction Algorithms

With the rapid growth of World Wide Web, efficient dimensionality reduction techniques have attracted much attention due to the need of preprocessing large-scale data or streaming data. Traditional FE approaches are not very practical for real-world tasks due to their high computational complexity. Although FS approaches are more efficient than the FE approaches, most of them are greedy and cannot provide the optimal solutions. In this section, according to the unified framework of dimensionality reduction techniques, we propose a highly scalable incremental FE algorithm and an optimal FS algorithm based on the Orthogonal Centroid algorithm. In Section 3.1, we propose a novel incremental FE algorithm, IOC. In Section 3.2, we propose a novel FS algorithm, OCFS. Then, in Section 3.3, we compare these two algorithms.

3.1 Incremental Orthogonal Centroid Algorithm

In this section, we propose the highly scalable incremental FE algorithm based on the OC algorithm. We call it the Incremental Orthogonal Centroid (IOC) algorithm.

3.1.1 Algorithm Derivation

Theorem 1 states that the traditional OC algorithm aims at optimizing $J(W) = trace(W^T S_b W)$ subject to $W \in H^{d\times p}$. Note that this optimization problem could be restated as $max \sum_{i=1}^{p} w_i S_b w_i^T$, subject to $w_i w_i^T = 1,\ i = 1, 2, \ldots, p$. We then introduce a Lagrange function as

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

where λ_k are the Lagrange multipliers. At the saddle point, the derivatives of L must vanish, leading to $S_b w_k^T = \lambda_k w_k^T$. Thus, w, the columns of W, are p leading eigenvectors of S_b. The traditional approach used to solve this problem is Singular Value Decomposition with high computational complexity. To make it applicable on large-scale data and to streaming data, we need a highly scalable incremental algorithm. To find the p leading eigenvectors of S_b incrementally, at the streaming data case, a sample sequence is presented as $\{x(n), l_n\}$ in this section, where $x(n)$ is the nth training data, and l_n is its corresponding class label, $n = 1, 2 \ldots$. A variable "S" at step n is denoted by "$S(n)$."

Lemma 1. *If* $\lim_{n\to\infty} a(n) = a$, *then* $\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} a(i) = a$.

This is a well-known lemma in mathematics. The Interclass scatter matrix after learning from the first n samples can be written as:

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

where $m_j(i)$ is the mean of class j at step i and $m(i)$ is the mean of training samples at step i. From the fact that $\lim_{n\to\infty} S_b(n) = S_b$ and Lemma 1, we obtain $S_b = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} S_b(i)$.

However, the general eigenvector form is $Au = \lambda u$, where u is the eigenvector of A corresponding to the eigenvalue λ. By replacing the matrix A with $S_b(n)$, we can obtain an approximate iterative eigenvector computation formulation with $v = \lambda u = Au$:

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

where $\Phi_j(i) = m_j(i) - m(i)$. Then, 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:

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

i.e.,

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

where $\alpha_j(n) = \Phi_j(n)^T v(n-1)/\|v(n-1)\|$, $j = 1, 2, \ldots, c$. For initialization, we set $v(0) = x(1)$.

Notice that eigenvectors are orthogonal to each other. Therefore, 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 jth eigenvector from the data,

$$x^{j+1}(n) = x^j(n) - (x^j(n)^T v^j(n))v^j(n)/\|v^j(n)\|^2,$$

where $x^1(n) = x(n)$. Since $m_l^j(n)$ and $m^j(n)$ are linear combinations of $x^j(i)$, where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, k$, and Φ_i are linear combinations of m_i and m, for convenience, we can only update Φ at each iteration step by,

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

In this way, the time-consuming orthonormalization process 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. The IOC algorithm summary is shown in Table 1. The solution of step n is $u^j(n) = v^j(n)/\|v^j(n)\|$, $j = 1, 2, \ldots, p$. Following the algorithm summary, we also give a simple example to illustrate how IOC solves the leading eigenvectors of S_b incrementally. Suppose the training data set is classified into two

TABLE 1
IOC Algorithm

for $n = 1, 2, \ldots$, do the following steps,
  $m(n) = ((n-1)m(n-1) + x(n))/n$
  $N_{l_n}(n) = N_{l_n}(n-1) + 1$
  $m_{l_n}(n) = (N_{l_n}(n-1)m_{l_n}(n-1) + x(n))/N_{l_n}(n)$
  $\Phi_i^1(n) = m_i(n) - m(n), i = 1, 2, \cdots, c$

  for $j = 1, 2, \ldots, \min\{p, n\}$
    if $j = n$ then
      $v^j(n) = x(n)$
    else
      $\alpha_i^j(n) = \Phi_i^j(n)^T v^j(n-1)/\|v^j(n-1)\|$
      $v^j(n) = \frac{n-1}{n}v^j(n-1) + \frac{1}{n}\sum_{i=1}^{c}\alpha_i^j(n)p_i(n)\Phi_i^j(n)$
      $\Phi_i^{j+1}(n) = \Phi_i^j(n) - \Phi_i^j(n)^T v^j(n)v^j(n)/\|v^j(n)\|\|v^j(n)\|$
    end if
  end for
end for

classes, represented by $a_i, i = 1, 2, \cdots$ and $b_i, i = 1, 2, \cdots$, respectively. Without loss of generality, we show the IOC algorithm on three leading samples a_1, a_2, b_1 of the training sequence. The initial values are given as: The initial mean $m(0), m_1(0), m_2(0)$ are all zero when no data in the stream arrive. We let the initial eigenvector $v^1(1) = a_1$ when the first data arrive.

At Step 1, the mean of all samples and the mean of Class 1 are both equal to a_1. The mean of Class 2 is still zero, i.e., $N_1(1) = 1, N_2(1) = 0, n = 1$ and $m_1(1) = a_1, m(1) = a_1$. Let $v^1(1) = a_1$ to denote the largest eigenvector at this step. There is no second eigenvector now due to the reason that there is only one training datum and the rank of S_b is at most 1.

At Step 2, when datum a_2 arrives, we have $N_1(2) = 2$, $N_2(2) = 0$, $n = 2$, $m_1(2) = (a_1 + a_2)/2, m_2(2) = 0$, and

$$m(2) = (a_1 + a_2)/2.$$

Then, we can compute $\Phi_1(2) = m_1(2) - m(2)$ and $\Phi_2(2) = m_2(2) - m(2)$. Let $\Phi_1^1(2) = \Phi_1(2)$, $\Phi_2^1(2) = \Phi_2(2)$, $\alpha_i^1(2) = \Phi_i^1(2)^T v^1(1)/\|v^1(1)\|^2$, and $i = 1, 2$. Then, the largest eigenvector can be updated by

$$v^1(2) = \frac{2-1}{2}v^1(1) + \frac{1}{2}\sum_{i=1}^{c}\alpha_i^1(2)\Phi_i^1(2).$$

It is a weighted linear combination of the latest eigenvector and the newly arrived data. Then, we can update,

$$\Phi_i^2(2) = \Phi_i^1(2) - \Phi_i^1(2)^T v^1(2)v^1(2)/\|v^1(2)\|^2, i = 1, 2.$$

The initial value of the second eigenvector is computed in an orthogonal space to the first eigenvector by $v^2(2) = a_2$. The solutions are $u^1(2) = v^1(2)/\|v^1(2)\|$ and $u^2(2) = v^2(2)/\|v^2(2)\|$.

At Step 3, we first update the sample number and means of different classes $N_1(3) = 2$, $N_2(3) = 1$, $n = 3$;

TABLE 2
Orthogonal Centroid Feature Selection

Step 1, compute the centroid m_i $i=1,2,...,c$ of each class for training data
Step 2, compute the centroid m of all training samples
Step 3, compute feature score $s(i)=\sum_{j=1}^{c}\frac{n_j}{n}(m_j^i-m^i)^2$ for all the features
Step 4, find the corresponding index set K consisted of the p largest ones in $S=\{s(i)\|1\le i\le d\}$

$m_1(3)=(a_1+a_2)/2$, $m_2(3)=b_1$, $m(3)=(a_1+a_2+b_1)/3$. Then, compute $\Phi_1^1(3)=\Phi_1(3)=m_1(3)-m(3)$,

$$\Phi_2^1(3)=\Phi_2(3)=m_2(3)-m(3),$$

and

$$\alpha_i^1(3)=\Phi_i^1(3)^T v^1(2)/\|v^1(2)\|^2, i=1,2.$$

The largest eigenvector could be updated by $v^1(3)=\frac{3-1}{3}v^1(2)+\frac{1}{3}\sum_{i=1}^{c}\alpha_i^1(3)\Phi_i^1(3)$. Then, we can solve the second eigenvector in an orthogonal space by

$$\Phi_i^2(3)=\Phi_i^1(3)-\Phi_i^1(3)^T v^1(3)v^1(3)/\|v^1(3)\|^2, i=1,2.$$
$$v^2(3)=\frac{2-1}{2}v^2(3)+\frac{1}{2}\sum_{i=1}^{c}\alpha_i^2(3)\Phi_i^2(3).$$

The solutions are $u^1(3)=v^1(3)/\|v^1(3)\|$ and

$$u^2(3)=v^2(3)/\|v^2(3)\|.$$

We can continue with these iterations until convergence is reached. Through the iteration steps, we can get the two converged vectors $u^1(n)$ and $u^2(n)$ at step n. We can consider these two vectors as column vectors of a matrix W, where W is the solution of the IOC Algorithm and it is an approximation of the batch Orthogonal Centroid algorithm. The convergence is identified by the proof summary in [15], [34].

3.1.2 Algorithm Analysis

Without loss of generality, suppose that $n >> p$. When new training data arrive, to solve the p leading eigenvectors, the algorithm needs p times of iterations, and each iteration step needs to compute c variables $\alpha_i^j(n), i=1,2,\cdots,c$. The main computational cost to solve each variable is the inner product between two d dimensional vectors. Thus, the time complexity of IOC when training a new input sample is $O(cdp)$, which is linear with each factor. It can be seen that it is applicable to process an online data stream based on moderate computational resources. However, when handling each input sample, IOC only needs keep the learned eigen-space and several first-order statistics of the past samples, such as the mean and the counts. The high-dimensional data are projected to the low-dimensional space one by one when the subspace is updated. Hence, the storage requirement is smaller compared to the batch techniques. Furthermore, IOC is a one-pass algorithm. If the dimensionality of data is very large which cannot be solved by batch algorithms, we can consider it as a data stream and compute the low-dimensional representation from the samples one by one. Due to these reasons, IOC is able to handle large-scale and a continuous data stream.

3.2 Orthogonal Centroid Feature Selection

Although some incremental FE algorithms have been proposed recently [1], [33], [34], to classify large-scale data or data streams such as Web documents, FS is still the most popularly used due to its efficiency. However, most of the current FS approaches are an approximation to the optimal solution in the solutions space according to some criteria. In this section, we propose a novel algorithm that can find the exact optimal solution according to the objective function of Orthogonal Centroid algorithm. We call it the Orthogonal Centroid Feature Selection (OCFS) algorithm.

From Theorem 1, the feature selection problem according to the objective function $J(\tilde{W})$ of OC algorithm is an optimization problem:

$$\arg\max J(\tilde{W})=\arg\max\ trace(\tilde{W}^T S_b\tilde{W})$$
$$\text{subject to } \tilde{W}\in\tilde{H}^{d\times p}.$$

Suppose $K=\{k_i, 1\le k_i\le d, i=1,2,\cdots,p\}$ is a group of indices of features. Since $\tilde{W}$ belongs to space $\tilde{H}^{d\times p}$, it must be a binary matrix with its elements of zero or one, and there is a unique nonzero element in each column. Following this constraint, let $\tilde{W}=\{\tilde{w}_i^k\}$ and let:

$$\tilde{w}_i^k=\begin{cases}1 & k=k_i\\ 0 & otherwise.\end{cases} \tag{1}$$

Then,

$$\begin{aligned} trace(\tilde{W}^T S_b\tilde{W}) &= \sum_{i=1}^{p}\tilde{w}_i{}^T S_b\tilde{w}_i \\ &= \sum_{i=1}^{p}\sum_{j=1}^{c}\frac{n_j}{n}(m_j^{k_i}-m^{k_i})^2. \end{aligned} \tag{2}$$

From (2), we can see that if a set of indices $K=\{k_i, 1\le k_i\le d, i=1,2,\cdots,p\}$ can maximize $\sum_{i=1}^{p}\sum_{j=1}^{c}\frac{n_j}{n}(m_j^{k_i}-m^{k_i})^2$, the binary matrix $\tilde{W}$ generated by K following (1) should maximize, $J(\tilde{W})=trace(\tilde{W}^T S_b\tilde{W})$. Then, this index set K should be the optimal solution of the feature selection problem according to the criterion $J(\tilde{W})$ subject to $\tilde{W}\in\tilde{H}^{d\times p}$. The problem now is to find an index set K such that $\sum_{j=1}^{c}\sum_{i=1}^{p}\frac{n_j}{n}(m_j^{k_i}-m^{k_i})^2$ is maximized. It can be seen that this could be solved simply by finding the p largest ones from $\sum_{j=1}^{c}\frac{n_j}{n}(m_j^k-m^k)^2, k=1,2,\cdots,d$. This motivates us to propose an optimal feature selection algorithm according to the criterion $J(\tilde{W})$. The details of the OCFS algorithm are given in Table 2.

From Table 2, the selected index set K can define a matrix $\tilde{W}$ by (1). This matrix is the solution of the optimization problem $J(\tilde{W})=\arg\max\ trace(\tilde{W}^T S_b\tilde{W})$ in

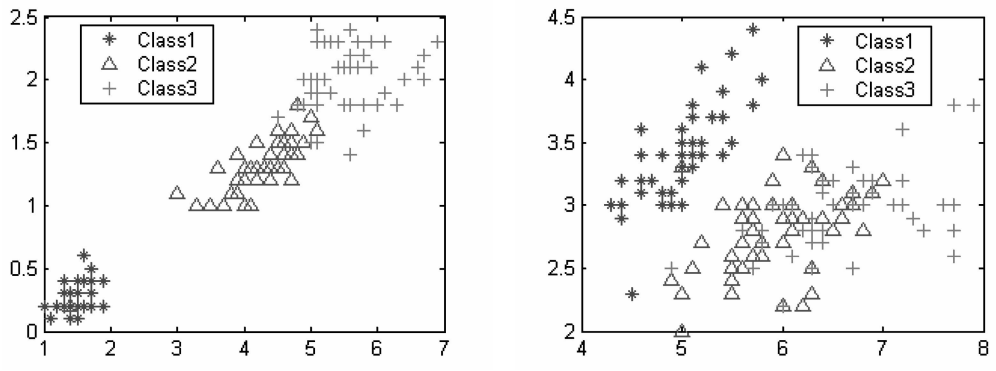


Fig. 2. Two-dimensional visualization of IRIS: (a) features picked by OCFS and (b) the two features left-out by OCFS.

the space $\tilde{W} \in \tilde{H}^{d\times p}$. We demonstrate our algorithm with a simple example. The UCI machine learning data set is a repository of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms.[1] We use the IRIS data set of UCI to show how our algorithm works. The documentation of this data set is complete, and there are three classes, four numeric attributes, and 150 samples. There are 50 samples in each class. Class 1 is linearly separable from the other two, but the other two are not linearly separable from each other. Without loss of generality and for intuition, we do not split the IRIS into training and testing data. Suppose $P = 2$, following our proposed OCFS: Step 1, computing the class mean of each class, respectively;

$$m_1 = \frac{1}{n_1}\sum_{x_i \in class\ 1} x_i = (5.006, 3.418, 1.464, 0.244),$$

$$m_2 = \frac{1}{n_2}\sum_{x_i \in class\ 2} x_i = (5.936, 2.770, 4.260, 1.326),$$

$$m_3 = \frac{1}{n_3}\sum_{x_i \in class\ 3} x_i = (6.588, 2.974, 5.552, 2.026).$$

Step 2, computing the mean of all the 150 samples;

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i = (5.8433, 3.054, 3.7587, 1.1987).$$

Step 3, computing the feature scores of all the features;

$$s(1) = \sum_{j=1}^{3}\frac{n_j}{n}(m_j^1 - m^1)^2 = 1.2642,$$

$$s(2) = \sum_{j=1}^{3}\frac{n_j}{n}(m_j^2 - m^2)^2 = 0.21955,$$

$$s(3) = \sum_{j=1}^{3}\frac{n_j}{n}(m_j^3 - m^3)^2 = 8.7329,$$

$$s(4) = \sum_{j=1}^{3}\frac{n_j}{n}(m_j^4 - m^4)^2 = 1.1621.$$

1. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Step 4, selecting the features corresponding to the indices of the two largest ones among $S = \{s(i)|1 \le i \le 4\}$. Then, represent the original data with these two features. It is clear that we should preserve the third and the first features here.

The OCFS aims at finding a group of features from all features such that this group of features can maximize $J(\tilde{W}) = trace(\tilde{W}^T S_b \tilde{W})$ in space $\tilde{W} \in H^{d\times p}$. Intuitively, OCFS aims at finding out a subset of features that can make the sum of distances between all the class means maximized in the selected subspace. Step 1 is to compute all the class means which could be used to represent different classes. Step 2 is to calculate the global mean and then the sum of distance among all the class means can be computed by computing the distance between each class mean and the global mean. In Step 3, the score of features which is the weighted sum of distance among all the class means along the direction of this feature are computed. Step 4 is used to select the directions with maximum sum of distance. Our theoretical analysis above can be used to prove that the features selected in this way are optimal according to our proposed criterion. Fig. 2 shows the two-dimensional visualization of IRIS by selecting two different features.

Fig. 2a is the IRIS in the two-dimensional space whose coordinates are selected by OCFS. Fig. 2b is the two-dimensional visualization of IRIS whose coordinates are the left-out features of OCFS. It can be seen that in the subspace selected by OCFS, the three classes are easier to be separated than in the other subspace.

The main computation time of OCFS is spent on the calculation of feature scores. Each feature score is calculated by the sum of c squared values. There are d feature scores to be calculated totally. Thus, its time complexity is $O(cd)$ which is the same as its counterparts: IG and CHI. However, OCFS only needs to compute the simple square function instead of functional computation such as logarithm of IG. Though the time complexity is the same, OCFS should be much more efficient. Experiments tell us that OCFS can process a data set with about half the time of what it takes for computing IG and CHI. OCFS is also robust since OCFS focuses only on the *means* of each class and all samples. That means that a little amount of mislabeled data cannot affect the final solution greatly.

TABLE 3
Comparison of the Two Proposed Algorithms

Aspects	IOC	OCFS
Categorization	Supervised feature extraction	Supervised feature selection
Rationale	Optimize a criterion that has the same solution as OC in a continuous space	Optimize a criterion that has the same solution as OC in a discrete space
Solution	Global optimal according to the given criterion	Global optimal in its discrete solution space according to the given criterion
Time complexity for a new sample arrive	$O(cdp)$	$O(cd)$
Time complexity for all training data	$O(cdpn)$	$O(cd)$
Property	Incremental algorithm	Efficient batch algorithm
Application area	Streaming data or large-scale data looked as streaming data	Large-scale data

3.3 Comparison of IOC and OCFS

Both proposed IOC and OCFS are supervised dimensionality reduction algorithms. We give an optimization model whose solutions are approximations to their counterparts of the Orthogonal Centroid algorithm in Section 2, and then we propose IOC and OCFS under this optimization model in Section 3. IOC is a feature extraction approach which aims at optimizing this criterion in a continuous solution space, while OCFS is a feature selection approach which aims at optimizing this criterion in a discrete solution space. Both of them can find the global optimal solution according to this criterion. However, the solution spaces are not the same. IOC, which is an incremental algorithm, can treat a large-scale data set as a data stream and process the high-dimensional data one by one without loading all the training data into memory. OCFS, on the other hand, is a batch algorithm and very efficient in processing very high-dimensional data with very low time complexity. Table 3 summarizes the detailed comparisons between these two methods for dimensionality reduction.

4 Experiments

In this section, we first describe the experimental settings which include the description of data sets, baseline algorithms, performance measurements, and key steps of experiments. And, then, we give the detailed experimental results. Finally, we use a section to discuss and analyze these experimental results.

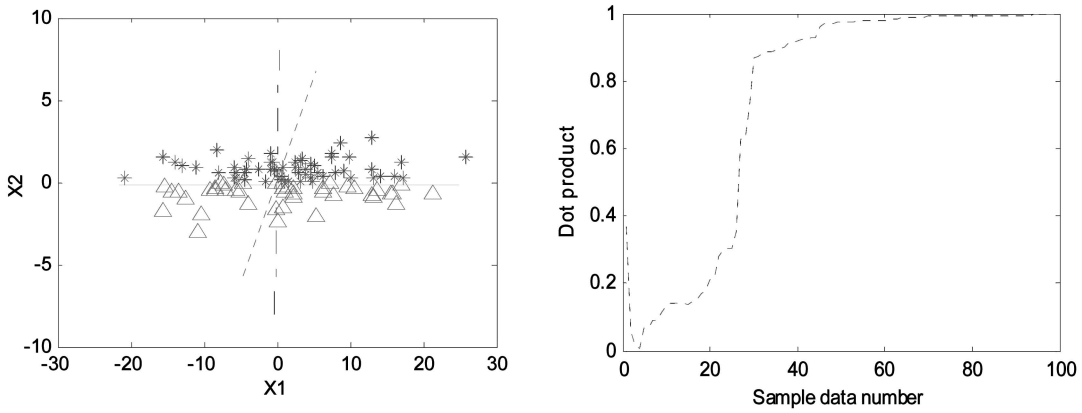
4.1 Experimental Setup

4.1.1 Data Sets

To show the performance of IOC and OCFS, we performed experiments on one synthetic data set for intuition and two real large-scale data sets: We use Reuters Corpus Volume 1 (RCV1) [16] and Open Directory Project (ODP)[2] which are two popularly used large-scale text data. For all the data sets that we use, we consider them as high-speed data streams, i.e., the samples are input one by one without any delay.

- **Synthetic Data**. We also generated the synthetic data by normal distribution. Fig. 3a shows a scatter plot of the data set. The stars are two-dimensional data points belong to Class 1, and the triangles are two-dimensional data points belong to Class 2. There are 100 training samples in a total of 50 samples in each of the two classes. We consider this as a two-dimensional data set as the data streams which are input to our algorithms one record at a time.
- **Reuters Corpus Volume 1**. Reuters Corpus Volume 1 (RCV1) data set which contains over 800,000 documents and the data dimension is about 500,000 by the traditional TFIDT indexing [28]. 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 randomly split them into five equal-sized subsets, and each time four of them are used as the training set and the remaining one is left as the test set. The experimental results reported in this paper are the average of the five runs. Moreover, we use this data set as a single label problem when training the classifier, i.e., we only keep the first label if a training sample is multilabeled.
- **Open Directory Project**. Open Directory Project (ODP) consists of Web documents crawled from the Internet. In this paper, we use the first layer ODP and only consider those documents in English and ignore all other non-English documents. Thirteen classes are used: Arts, Business, Computers, Games, Health, Home, Kids and Teens, News, Recreation,

2. http://rdf.dmoz.org/.



(a) (b)

Fig. 3. Results on synthetic data. (a) Reduced space on synthetic data. (b) Convergence curve of IOC on synthetic data.

Science, Shopping, Society, and Sports. The TFIDF indexing is also used here to represent the text data as a real valued term by document matrix. There are 1,585,007 documents and each document is a 966,410-dimensional vector. We also randomly split them into five equal-sized subsets, and each time four of them are used as the training set and the remaining one is left as the test set.

4.1.2 Baseline Algorithms

For the FE algorithms, we used linear approaches introduced above, PCA, LDA, and MMC, as baseline algorithms no matter whether they are supervised or unsupervised. Since all the batch approaches fail on the large-scale data sets, the baselines are conducted by their corresponding incremental versions, i.e., Incremental PCA [33], Incremental LDA [8], and Incremental MMC [34].

There are many FS algorithms for data preprocessing of classification problems. Among them, Information Gain (IG) and χ^2-test (CHI) are dominant in the area of text categorization since they have been proven to be very effective and efficient [5], [36]. Moreover, they are two of the most widely used dimensionality reduction algorithms for real Web document categorization problems [36]. Thus, in this paper, we chose the IG and CHI as the baseline feature selection algorithms.

IG and CHI are the state-of-the-art feature selection approaches. In this paper, we applied them on all training data to generate 10, 100, 1,000, and 10,000-dimensional spaces. However, the original IG and CHI cannot select a given number of features globally; it selects a given number of features for each class. Moreover, there are always a number of overlapping features selected from different classes. Thus, it is very difficult to control the global number of features selected by IG and CHI. To solve this problem, we selected the given number of features by computing their average score in different classes and select the largest ones to meet the given number.

4.1.3 Performance Measurement

Precision, Recall, and F1 are the most widely used performance measurements for text categorization problems nowadays. Precision is the ratio of the number of correctly categorized data to the number of all testing data. Recall is the ratio of the number of correctly categorized data to the number of all the assigned data. F1 is a popular measure in text categorization that combines recall and precision. In this paper, we use Micro F1 measure as our effectiveness measurement which combines recall and precision into a single score according to the following formula: Micro $F1 = \frac{2P \times R}{P+R}$, where P is the Precision and R is the Recall. In all figures, we use F1 to denote Micro F1. Note that we ignore the results of Macro F1 which is also a popular metric since it can get similar conclusion with Micro F1 in our experiments.

Efficiency is evaluated by the CPU runtime. We used a computer with Pentium(R) 4 CPU 2.80GHz, 1GB of RAM to conduct the experiments. The programming language used is C++ 7.0. To the convergence of incremental FE approaches, inner product was used. Since for vectors v and v' we have $\|v - v'\| = 2(1 - v \cdot v')$, and $v = v'$ iff $v \cdot v' = 1$, the correlation between two unit eigenvectors is represented by their inner product. The larger the inner product, the more similar the two eigenvectors are. The solutions of the incremental algorithms are the approximation of their corresponding batch algorithms. We measure the convergence performance by computing the inner product between the learned eigenvector at each iteration step and the target eigenvector solved by the corresponding batch algorithm.

4.1.4 Key Steps of Experiments on Large-Scale Data

We conducted all the dimensionality reduction algorithms involved on the synthetic data for intuition and conduct all the involved algorithms on RCV1 data to compare the performance of FE and FS approaches. Since the FS approaches are much more popular than FE approaches on real large-scale data due to their efficiency, we give some additional experiments for feature selection on ODP whose scale is even larger than RCV1. The experiments consist of the following steps:

- applying the dimensionality reduction algorithm on a specific size of training data to learn a low dimensional presentation,

TABLE 4
Dimensionality Reduction on RCV1 for Classification

	IPCA3	IMMC3	ILDA3	IOC3	IG3	CHI3	OCFS3
F1	0.4806	0.8063	***0.8169***	***0.8169***	0.4475	0.4475	0.4475
Time (s)	22,960.17	48,813.51	70,574.02	26,884.6	66.72	68.83	***35.96***
Converge step	***1,362***	4,657	10,023	1,776			

- transforming all the training data into the learned low-dimensional space,
- training SVM by SMO [24] using the transformed low-dimensional training data,
- transforming all the test data to the low-dimensional space, and
- evaluating the classification performance, using F1 value, on the transformed test data.

We use the approaches below to select the dimension of the reduced space:

- Since the dimension of the reduced data space solved by LDA is limited by the class number ($c-1$), only a three-dimensional subspace is available on RCV1. To compare the performance, we use all the involved algorithms to generate a three-dimensional subspace on RCV1. They are denoted by IOC3, IG3, and LDA3, etc.
- To show the performance of feature selection approaches on different scales of reduced space, we use them to reduce the dimension of RCV1 and ODP to 10, 100, 1,000, and 10,000, respectively.

4.2 Experimental Results

4.2.1 Synthetic Data

In Fig. 3a, we show the directions of one-dimensional space of the original data solved by OCFS, IOC, OC, PCA, LDA, and MMC in the same figure. The one-dimensional spaces of OCFS, LDA, and MMC are overlapped and plotted by the vertical dashed line. The one-dimensional space of PCA is plotted by horizontal straight line. The one-dimensional spaces solved by IOC and batch OC are overlapped and denoted by the gradient dotted line. It can be seen that if we project all the samples to the one-dimensional space learned by the unsupervised PCA algorithm, the data of two classes mix since PCA ignores the valuable label information in the training data. On the other hand, our two proposed supervised algorithms could make the data of different classes easier to be separated in the learned one-dimensional space. Moreover, the IOC algorithm could approximate the solution of batch OC algorithm.

Fig. 3b shows the curve of inner product between the solution vector solved by batch Orthogonal Centroid algorithm and the solution vector solved by our IOC at each iteration step. In other words, it is the convergence curve of our proposed IOC on this small toy data. It can be seen that IOC can converge after learning from about 40 training samples.

4.2.2 RCV1 Data

As demonstrated above, we reduce the dimension of RCV1 data to a three-dimensional space by IPCA, IMMC, ILDA, IG, CHI, OCFS, and IOC, respectively. The results are listed in Table 4. It is hard to give the convergence curve by inner product since we cannot calculate the target subspaces by the batch algorithms on such a large-scale data set. However, we can draw the conclusion that an algorithm has converged through another way. In other words, if the summation of Euclidean distance between the column vectors of the projection matrix at step t and step $t-1$ is smaller than a given threshold (we use 0.06 in this paper), we stop our iteration. Table 4 also gives the average convergence steps of these algorithms.

Since the FS approaches are much more efficient than the FE approaches, they are more often used in real large-scale data tasks nowadays. To additionally capture the performance of FS approaches, Fig. 4 gives the performance of OCFS in contrast to IG and CHI. Fig. 4a shows the F1 value with different numbers of selected features and Fig. 4b shows the real CPU runtime.

4.2.3 ODP Data

The performance of different FS algorithms on ODP data with CPU runtime are reported in Fig. 5. Fig. 5a shows the F1 curve with different number of selected features and Fig. 5b shows the CPU runtime.

Since the ODP is very large-scale sparse data, too low dimension such as 10 could make most of its samples to be zero vectors no matter which feature selection approach is used. Thus, we ignore the 10-dimensional space on ODP. Instead, to show the results in low-dimensional space, we add the performance in 200-dimensional space and 500-dimensional space in this experiment. Due to the scale

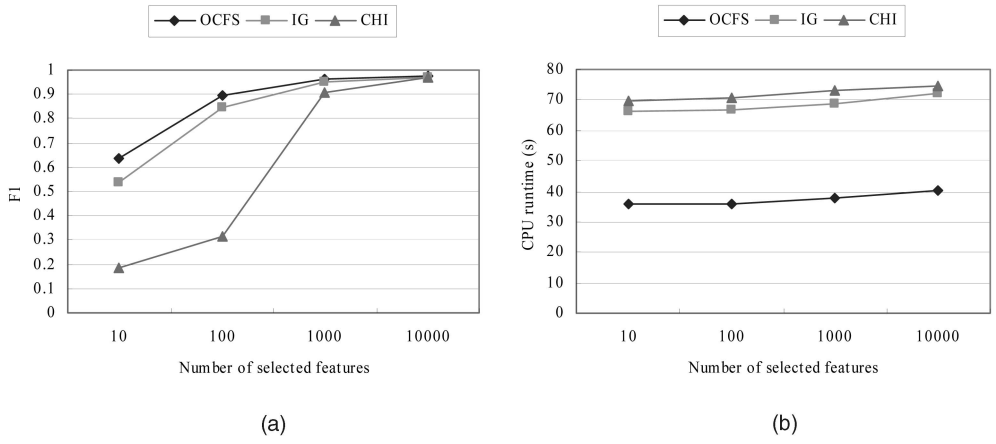


Fig. 4. Performance of feature selection algorithms on RCV1 data. (a) F1 measure for classification. (b) CPU runtime.

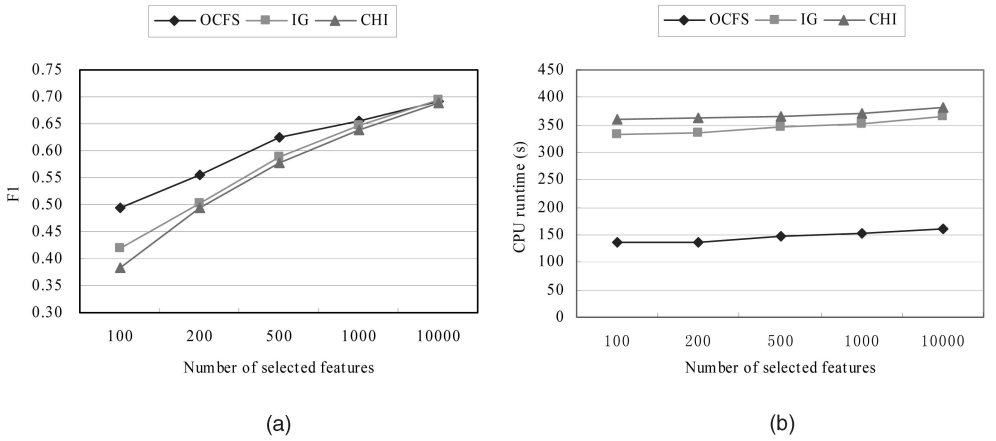


Fig. 5. Performance of feature selection algorithms on ODP data. (a) F1 measure for classification. (b) CPU runtime.

of the data, ILDA failed on this data set in our experiments. The parameter-learning problem of IMMC makes this algorithm unsatisfactory. In contrast to the unsupervised IPCA, as an example, IOC can achieve 57.2 percent F1 improvement than IPCA in a 15-dimensional reduced space. From the experimental results, we can see that the CPU time of OCFS is less than half of its counterpart of IG and CHI. The F1 measurements of OCFS are always better than IG and CHI.

4.3 Discussion and Analysis of Experimental Results

For the *Feature Extraction* algorithms, from the experiments (Table 4), we can see that IOC can outperform others for classification problems. Though IPCA is more efficient than IOC, it is unsupervised and ignores the valuable label information for classification. Thus, it cannot get comparable performance with IOC. Though the ILDA can get the same performance with IOC, its efficiency is not comparable. As a trade-off the IOC outperforms other FE approaches in both efficiency and effectiveness.

For the *Feature Selection* algorithms, from the experiments, we can see that the proposed OCFS is consistently better than IG and CHI especially when the reduced dimension is extremely small for large-scale data categorization problems. At the same time, it is more efficient than the others by using only about half of the time used by baselines to select good features. For very large-scale data such as the rapidly growing Web data, saving about half of the computational time is valuable and important. From the dimension by Micro F1 figures, we can draw the conclusion that OCFS can get significant improvements over baselines when the selected subspace dimension is extremely small while we get slightly better performance when the selected subspace dimension is relatively large. This phenomenon occurs due to the reason that when the selected feature dimension is small, the proposed OCFS, which is an optimal FS approach, can outperform the greedy ones. With the increasing number of selected features, the saturation of features makes additional features of less value. When the number of selected features is large enough, all FS algorithms involved can achieve comparable performance, no matter if they are optimal or greedy.

5 Conclusions and Future Work

In this paper, we proposed two effective and efficient dimensionality reduction algorithms for data preprocessing of high-dimensional data and streaming data classification problems. These two algorithms are based on the Orthogonal Centroid algorithm. We first reformulated the Orthogonal Centroid algorithm into a constrained optimization problem. The IOC is then designed to solve a challenging issue of computing the dominating eigenvectors from incrementally arriving sample streams without storing the

previously received data in advance. As a result, the algorithms scan through the data only once. The OCFS is an optimal feature selection approach designed according to the objective function of OC. These two algorithms are related to two different but related types of dimensionality reduction approaches: The IOC algorithm addresses the *Feature Extraction* problem and successfully overcomes the high-complexity issue for large-scale, online learning, whereas the OCFS algorithm addresses the *Feature Selection* problem and ensures the closed form optimal solution of the objective function from Orthogonal Centroid algorithm. Although most dimensionality reduction approaches break down on large-scale streaming Web documents, our proposed algorithms can be conducted effectively and efficiently. We also discussed the relationship between FE approaches and FS approaches under a unified framework which could help the readers choose other suitable dimensionality reduction algorithms for classification tasks.

From the proposed framework for dimensionality reduction, we can see that dimensionality reduction could be treated as the optimization of some objective function. FE is the optimization in a continuous solution space and FS is the optimization in a discrete solution space. Thus, just like IOC and OCFS, each FE algorithm in continuous solution space should correspond to a FS algorithm in discrete space and vise versa. In the future, we plan to extend our algorithm into more complex dimensionality reduction algorithms such as optimal feature selection by the Maximum Margin Criterion. Moreover, more experiments on real tasks such as online search result-reduction and classification are to be conducted.

ACKNOWLEDGMENTS

Qiang Yang is supported by a grant from Hong Kong RGC (HKUST 6187/04E).

REFERENCES

[1] M. Artae, M. Jogan, and A. Leonardis, "Incremental PCA for On-Line Visual Learning and Recognition," *Proc. 16th Int'l Conf. Pattern Recognition,* pp. 781-784, 2002.
[2] A.L. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," *Artificial Intelligence,* vol. 97, nos. 1-2, pp. 245-271, 1997.
[3] M. Belkin and P. Niyogi, "Using Manifold Structure for Partially Labelled Classification," *Proc. Conf. Advances in Neural Information Processing,* pp. 929-936, 2002.
[4] S.E. Brian and G. Dunn, *Applied Multivariate Data Analysis.* Edward Arnold, 2001.
[5] K. Daphne and M. Sahami, "Toward Optimal Feature Selection," *Proc. 13th Int'l Conf. Machine Learning,* pp. 284-292, 1996.
[6] W. Fan, M.D. Gordon, and P. Pathak, "Effective Profiling Of Consumer Information Retrieval Needs: A Unified Framework And Empirical Comparison," *Decision Support Systems,* vol. 40, pp. 213-233, 2004.
[7] J.E. Gentle, *Numerical Linear Algebra for Applications in Statistics.* Springer-Verlag, 1998.
[8] K. Hiraoka, K. Hidai, M. Hamahira, H. Mizoguchi, T. Mishima, and S. Yoshizawa, "Successive Learning of Linear Discriminant Analysis: Sanger-Type Algorithm," *Proc. 14th Int'l Conf. Pattern Recognition,* pp. 2664-2667, 2000.
[9] R. Hoch, "Using IR Techniques for Text Classification in Document Analysis," *Proc. 17th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 31-40, 1994.
[10] P. Howland and H. Park, "Generalizing Discriminant Analysis Using the Generalized Singular Value Decomposition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, pp. 995-1006, 2004.
[11] M. Jeon, H. Park, and J.B. Rosen, "Dimension Reduction Based on Centroids and Least Squares for Efficient Processing of Text Data," Technical Report MN TR 01-010, Univ. of Minnesota, Minneapolis, Feb. 2001.
[12] I.T. Jolliffe, *Principal Component Analysis.* Springer-Verlag, 1986.
[13] J.F. Hair, R.L. Tatham, R.E. Anderson, and W. Black, *Multivariate Data Analysis,* fifth ed. Prentice Hall, Mar. 1998.
[14] R. Kohavi and G. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence,* vol. 97, nos. 1-2, pp. 273-324, 1997.
[15] H.J. Kushner and D.S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* New York: Springer-Verlag, 1978.
[16] D. Lewis, Y. Yang, T. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *J. Machine Learning Research,* pp. 361-397, 2003.
[17] D.D. Lewis, "Feature Selection and Feature Extraction for Text Categorization," *Proc. Workshop Speech and Natural Language,* pp. 212-217, 1992.
[18] H. Li, T. Jiang, and K. Zhang, "Efficient and Robust Feature Extraction by Maximum Margin Criterion," *Proc. Conf. Advances in Neural Information Processing Systems,* pp. 97-104, 2004.
[19] Y. Li, L. Xu, J. Morphett, and R. Jacobs, "An Integrated Algorithm of Incremental and Robust PCA," *Proc. Int'l Conf. Image Processing,* pp. 245-248, 2003.
[20] R.-L. Liu and Y.-L. Lu, "Incremental Context Mining for Adaptive Document Classification," *Proc. Eighth ACM Int'l Conf. Knowledge Discovery and Data Mining,* pp. 599-604, 2002.
[21] A.M. Martinez and A.C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, pp. 228-233, 2001.
[22] E. Oja, "Subspace Methods of Pattern Recognition," *Pattern Recognition and Image Processing Series,* vol. 6, 1983.
[23] H. Park, M. Jeon, and J. Rosen, "Lower Dimensional Representation of Text Data Based on Centroids and Least Squares," *BIT Numerical Math.,* vol. 43, pp. 427-448, 2003.
[24] J. Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization," *Advances in Kernel Methods: Support Vector Learning,* pp. 185-208, 1999.
[25] B.-Y. Ricardo and R.-N. Berthier, *Modern Information Retrieval.* Addison Wesley Longman, 1999.
[26] R.O. Duda, P.E. Hart, and D.G Stork, *Pattern Classification,* second ed. John Wiley, 2001.
[27] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, pp. 2323-2326, 2000.
[28] G. Salton and C. Buckley, "Term Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management,* vol. 24, pp. 513-523, 1988.
[29] M. Spitters, "Comparing Feature Sets for Learning Text Categorization," *Proc. Int'l Conf. Computer-Assisted Information Retrieval,* pp. 233-251, 2000.
[30] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, pp. 2319-2323, 2000.
[31] R.J. Vaccaro, *SVD and Signal Processing II: Algorithms, Analysis and Applications.* Elsevier Science, 1991.
[32] A.R. Webb, *Statistical Pattern Recognition,* second ed. John Wiley, 2002.
[33] J. Weng, Y. Zhang, and W.-S. Hwang, "Candid Covariance-Free Incremental Principal Component Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, pp. 1034-1040, 2003.
[34] J. Yan, B.Y. Zhang, S.C. Yan, Z. Chen, W.G. Fan, Q. Yang, W.Y. Ma, and Q.S. Cheng, "IMMC: Incremental Maximum, Marginal Criterion," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 725-730, 2004.
[35] J. Yan, N. Liu, B.Y. Zhang, S.C. Yan, Q.S. Cheng, W.G. Fan, Z. Chen, W.S. Xi, and W.Y. Ma, "OCFS: Orthogonal Centroid Feature Selection," *Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* 2005.
[36] Y. Yang and J.O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," *Proc. 14th Int'l Conf. Machine Learning,* pp. 412-420, 1997.

Jun Yan received the bachelor's degree from the Mathematical Science Department at Jilin University in 2001. He is a PhD student in the Department of Information Science, School of Mathematical Science, Peking University, P.R. China. His major is digital signal processing and pattern recognition. He has worked at Microsoft Research Asia as an intern since November 2004 and his previous research works focused on Web search and mining. Currently, he is a research associate at CBI, HMS, Harvard.

Benyu Zhang received the bachelor's and master's degrees in computer science from Peking University in 1999 and 2002. He joined Microsoft Research, China, in July 2002 to pursue his wide-ranging research interests in machine learning, information retrieval, data mining, and artificial intelligence.

Ning Liu received the bachelor's degree from the Mathematical Science Department at Jilin University in 2001. She is a PhD student at the School of Mathematical Sciences, Tsinghua University, China. She has also been an intern at Microsoft Research Asia since February 2004. Her research interests include data mining and information retrieval.

Shuicheng Yan received the BS and PhD degrees both from the Applied Mathematics Department, School of Mathematical Sciences, Peking University, P.R. China in 1999 and 2004, respectively. His research interests include computer vision and machine learning.

Qiansheng Cheng received the BS degree in mathematics from Peking University, Beijing, China, in 1963. He is now a professor in the Department of Information Science, School of Mathematical Sciences, Peking University, China, and he was the vice director of the Institute of Mathematics, Peking University, from 1988 to 1996. His current research interests include signal processing, time series analysis, system identification, and pattern recognition. He is the vice chairman of Chinese Signal Processing Society and has won the Chinese National Natural Science Award.

Weiguo Fan received the PhD degree in Information Systems from the University of Michigan Business School, Ann Arbor, in July 2002, the MS degree in computer science from the National University of Singapore in 1997, and the BE degree in information and control engineering from the Xi'an Jiaotong University, China, in 1995. He is an assistant professor of information systems and computer science at the Virginia Polytechnic Institute and State University. His research interests include information retrieval, data mining, and text/Web mining.

Qiang Yang received the bachelor's degree from Peking University in China in 1982 and the PhD degree from University of Maryland, College Park, in 1989. He was a faculty member at the University of Waterloo and Simon Fraser University in Canada between 1989 and 2001. At Simon Fraser University, he held an NSERC Industrial Chair from 1995 to 1999. He is currently a faculty member at Hong Kong University of Science and Technology. His research interest is in artificial intelligence and data mining.

Wensi Xi received the bachelor's degree from Shanghai Jiaotong University, China, 1997 and the master's degrees from Nanyang Technological University, Singapore, 2000. He is currently a PhD candidate at the Computer Science Department, Virginia Polytechnic Institute and State University, Blacksburg. His research topic is "information integration using Unified Relationship Matrix."

Zheng Chen received the bachelor's, master's, and PhD engineering degrees in computer science at Tsinghua University in 1994 and 1999. He joined Microsoft Research, China, in March 1999 to pursue his wide-ranging research interests in machine learning, information retrieval, speech recognition, natural language processing, multimedia information retrieval, personal information management, and artificial intelligence.