Incremental Kernel Null Space Discriminant Analysis for Novelty Detection

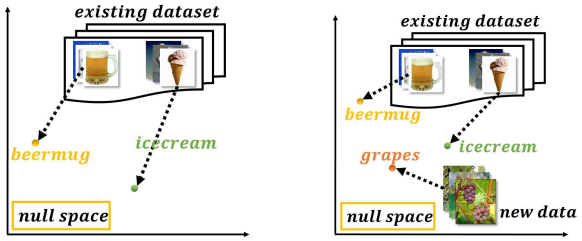
Juncheng Liu[1], Zhouhui Lian[1]*, Yi Wang[2], Jianguo Xiao[1]
[1]Institute of Computer Science and Technology, Peking University, China
[2]Dalian University of Technology
Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China

Abstract

Novelty detection, which aims to determine whether a given data belongs to any category of training data or not, is considered to be an important and challenging problem in areas of Pattern Recognition, Machine Learning, etc. Recently, kernel null space method (KNDA) was reported to have state-of-the-art performance in novelty detection. However, KNDA is hard to scale up because of its high computational cost. With the ever-increasing size of data, accelerating the implementing speed of KNDA is desired and critical. Moreover, it becomes incapable when there exist successively injected data. To address these issues, we propose the Incremental Kernel Null Space based Discriminant Analysis (IKNDA) algorithm. The key idea is to extract new information brought by newly-added samples and integrate it with the existing model by an efficient updating scheme. Experiments conducted on two publicly-available datasets demonstrate that the proposed IKNDA yields comparable performance as the batch KNDA yet significantly reduces the computational complexity, and our IKNDA based novelty detection methods markedly outperform approaches using deep neural network (DNN) classifiers. This validates the superiority of our IKNDA against the state of the art in novelty detection for large-scale data.

1. Introduction

Novelty detection, which aims to identify new or unknown data that a system has not been trained with and was not previously aware of [17], is a fundamental and on-going research problem in areas of Pattern Recognition, Machine Learning, Computer Vision, etc [21]. The novelty detection procedure can be regarded as a binary classification task where the positive exemplars are available and the negative ones are insufficient or absent. To be a good classification system, the ability to differentiate between known and unknown objects during testing is desired. Yet, the problem is tough since only the statistics of the already known information can be utilized. Novelty detection finds many real applications in our daily lives. For animals, novelty probably means a potential predator or a threat. Recognizing these unusual objects swiftly could survive themselves. Novelty detection also has practical significance in medical diagnoses. Doctors always need to pick up abnormality in a patient's index which has large possibility to be diagnosed as a disease.



(a) Null space projections (b) Incremental update

Figure 1: An illustration of our IKNDA algorithm. Each class is projected into a single point in the joint null space as illustrated in (a). During updating stage, our approach updates the null space basis from novel classes (grapes in this example) using the proposed incremental null space method, taking advantage of previous information. Adding new class is equivalent to updating and adding a single point in this subspace as illustrated in (b).

However, existing classification methods typically neglect the importance of this property. Most of these advanced classifiers including deep neural network (DNN) based approaches that are widely used today make the assumption of "closed world", in which the categories of testing instances should not be out of the range of training classes. This rarely holds in real cases since many important classes might be under-represented and some classes might not be included in the training set. To solve this problem, a number of novelty detection methods have been proposed. For a thorough review please refer to [18, 21]. In

*Corresponding author. E-mail: lianzhouhui@pku.edu.cn

2013, Bodesheim *et al.* [2] presented a novelty detection method by leveraging the kernel null space based discriminant analysis which reports state-of-the-art performance on two datasets. Their algorithm maps all instances within one class to a single point in a joint null space, the novelty score is obtained by simply calculating the shortest distance between the instance and the mapped classes. However, the algorithm is hard to scale up because of the intensive computation burden brought by the eigen-decomposition of the kernel matrix. Additionally, their method cannot handle the database where new data are injected successively, which is a typical characteristic in novelty detection tasks. To address these issues, we propose an incremental version of the kernel null space discriminant analysis. Our algorithm is able to handle incremental recognition tasks while significantly reduces computing time without sacrificing detection accuracy.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 briefly describes the basic concepts of null space based LDA and its kernelization. Then, our Incremental Kernel Null Space based Discriminant Analysis (IKNDA) algorithm is presented with detailed mathematical analyses in Section 4. Computational complexity and storage requirement are analyzed in Section 5. Section 6 presents experimental results, and Section 7 concludes this paper by amplifying advantages of our method and pointing out future work.

2. Related Work

As a subspace learning method, Linear Discriminant Analysis (LDA) [14] and its variations have been studied for many years. They have been widely applied in many applications of Pattern Recognition, Computer Vision, etc. such as face recognition [5, 16], text-image combination multimedia retrieval [19], speech and music classification [1], outliers detection [22], generalized image and video classification [20, 24], and so on. A crucial step of these algorithms lies in eigen-decomposition, which has a complexity of $\boldsymbol{O}(N^3)$. The computational time increases sharply as the scale of training dataset enlarges. Using existing approaches, during on-line updating process, new samples are successively added to the existing training set, which makes the batch computation quite inefficient. To solve this problem, many efficient incremental algorithms have been proposed [20, 9, 11, 27, 7]. Meanwhile, Kim *et al.* applied the concept of sufficient spanning set approximation in each updating step [12]. Their algorithm reduced the complexity to $\boldsymbol{O}(md^2)$ where m and d denote the sample feature dimension and the reduced subspace dimension, respectively. Sharma *et al.* proposed a fast implementation of null space LDA using random matrix multiplication [23]. However, these methods only consider the linear feature space, whereas kernel induced feature space is more suitable for data with highly complex and non-linear distributions [6]. Latter, Xiong *et al.* [25] proposed a QR decomposition-based KDA algorithm in which QR decomposition is applied rather than eigen-decomposition. However, it is hard to convert this algorithm to incremental learning mode, which hinders the further improvement of the method's performance. In 2007, a spectral regression-based KDA method was presented by Cai *et al.* [4], it is shown that their algorithm is 27 times faster than the ordinary KDA. This paper proposes a new method that has even lower computational cost and provides an intuitive way to conduct incremental learning.

One major restriction when using LDA is the non-singularity of within-class scatter matrix must be guaranteed, which is not always satisfied in practical situations. To overcome this limitation, Yu *et al.* [26] and Huang *et al.* [10] proposed the null space based method (KNDA) which takes advantage of the null space [16]. The null space method is suitable for the class-incremental process due to its inherent nature, i.e., each class is projected into one single point in the null space where within-class scatter vanishes and between-class scatter remains. Therefore, adding a new class is equivalent to updating and adding a single point in the subspace as illustrated in Figure 1.

Recently, Bodesheim *et al.* [2] reported that KNDA outperforms other existing methods in novelty detection. However, as mentioned above, batch KNDA is hard to scale up because of high computational cost and large storage requirement, which limits the scenarios where KNDA can be applied.

Developing an incremental method for kernel DAs is relatively more difficult than linear DAs. In linear case, the novel basis perpendicular to the space spanned by existing centralized samples is expected to be relatively rare. Especially when the amount of samples grows to reach the dimensionality of features, no new information will be introduced (same as the term low-rank). The above-mentioned assumption is critical for many existing updating algorithms such as [3]. However, in kernel induced feature space, the situation is quite different, i.e., the number of bases in this space can be larger than the dimension of feature vectors, which makes it impossible for these existing algorithms to work properly.

In our IKNDA, new information is extracted from newly-added samples and the proposed incremental null space is employed. As we know, the null space based LDA typically suffers from the over-fitting of training data, which however only has slight influence in our method compared to the Kernel LDA. Furthermore, in contrast to other existing methods, our approach takes advantage of a joint null space (shown in Figure 1). Experimental results demonstrate that the proposed IKNDA method yields performance similar as that of the batch KNDA approach yet significantly re-

Notation	Descriptions
D	Dimensionality of features
N	Number of training data
$\mathbf{X}$	Existing data matrix
$\mathbf{Y}$	Newly-injected data matrix
$\mathbf{Z}$	Updated data matrix
$\mathbf{B}$	Orthogonal basis of zero-mean data
$\mathbf{\Pi}$	Centralizing operator by class-wise mean
$\mathbf{H}$	Centralizing operator by global mean
$\mathbf{V}$	Coefficients of $\mathbf{X}_\Phi$ for constructing $\mathbf{B}$
$\boldsymbol{\beta}$	Null space of $\mathbf{D}$
$\mathbf{X}_c$,$\mathbf{Y}_c$,$\mathbf{Z}_c$	Centralized data matrix
$\mathbf{X}_\Phi$,$\mathbf{Y}_\Phi$,$\mathbf{Z}_\Phi$	Matrices in kernel induced feature space
$\boldsymbol{\mu}_X$,$\boldsymbol{\mu}_Y$,$\boldsymbol{\mu}_Z$	Mean vectors of $\mathbf{X}$,$\mathbf{Y}$ and $\mathbf{Z}$, respectively
$\mathbf{K}$,$\mathbf{K}_1$,$\mathbf{K}_2$	Kernel matrices: $\mathbf{X}_\Phi^T\mathbf{X}_\Phi$,$\mathbf{X}_\Phi^T\mathbf{Y}_\Phi$,$\mathbf{Y}_\Phi^T\mathbf{Y}_\Phi$

Table 1: Notations

duces the computational complexity (more than 100 times faster in some cases) in our experiments carried out on two publicly-available datasets. By taking advantage of our proposed algorithm, the kernel null space based LDA can be more widely applied especially when handling large-scale and incremental learning problems.

3. Mathematical Background

Given $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in \mathcal{R}^{D\times N}$, the main idea is to minimize the within-class scatter and maximize the between-class scatter simultaneously. Let $\mathbf{S}_b$ denote the between-class scatter matrix and $\mathbf{S}_w$ denote the within-class scatter matrix. LDA aims at finding a subspace basis $\boldsymbol{\varphi}$ which maximizes the so-called fisher criterion as:

$$\mathbf{J}(\boldsymbol{\varphi}) = \frac{\boldsymbol{\varphi}^T\mathbf{S}_b\boldsymbol{\varphi}}{\boldsymbol{\varphi}^T\mathbf{S}_w\boldsymbol{\varphi}} \quad . \tag{1}$$

It is well known that the solution is given by solving a generalized eigenvalue problem:

$$\mathbf{S}_b\boldsymbol{\varphi} = \lambda\mathbf{S}_w\boldsymbol{\varphi} \quad . \tag{2}$$

The eigenvectors $\boldsymbol{\varphi}^{(1)}, \boldsymbol{\varphi}^{(2)}, \ldots, \boldsymbol{\varphi}^{(k)}$ corresponding to k largest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ are collected as columns of a transform matrix $\mathbf{A}$ and discriminative features of LDA are computed by:

$$\boldsymbol{y}_i = \mathbf{A}^T\boldsymbol{x}_i, \ \ \forall i = 1, 2, \ldots, N. \tag{3}$$

The null space method is considered as a special case of LDA when the within-class scatter in subspace vanishes and the between-class scatter remains, which is also known as Null Foley Sammon Transform(NFST) [8]:

$$\begin{cases} \boldsymbol{\varphi}^T\mathbf{S}_w\boldsymbol{\varphi} = 0 \\ \boldsymbol{\varphi}^T\mathbf{S}_b\boldsymbol{\varphi} > 0 \end{cases} , \tag{4}$$

and it is equivalent to:

$$\begin{cases} \boldsymbol{\varphi}^T\mathbf{S}_t\boldsymbol{\varphi} > 0 \\ \boldsymbol{\varphi}^T\mathbf{S}_w\boldsymbol{\varphi} = 0 \end{cases} , \tag{5}$$

where $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$ denotes the total scatter matrix. It can be observed that the projections lie in the null space of $\mathbf{S}_w$. Let $\mathbf{N}_t$ and $\mathbf{N}_w$ be the null space of $\mathbf{S}_t$ and $\mathbf{S}_w$, respectively, and $\mathbf{N}_t^\perp, \mathbf{N}_w^\perp$ denoting their orthogonal complements, we have:

$$\begin{aligned} \mathbf{N}_t = &\quad \{\mathbf{z} \in \mathcal{R}^D | \mathbf{S}_t\mathbf{z} = 0\} \\ \mathbf{N}_w = &\quad \{\mathbf{z} \in \mathcal{R}^D | \mathbf{S}_w\mathbf{z} = 0\} \quad . \end{aligned} \tag{6}$$

It is easy to verify that the projection $\boldsymbol{\varphi}$ lies in the space:

$$\boldsymbol{\varphi} \in \mathbf{N}_t^\perp \cap \mathbf{N}_w \quad . \tag{7}$$

Then $\boldsymbol{\varphi}$ can be represented by a set of orthogonal bases $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ of $\mathbf{N}_t^\perp$ as:

$$\boldsymbol{\varphi} = \mathbf{B}\boldsymbol{\beta} \quad . \tag{8}$$

It has been proved [8] that $\mathbf{N}_t^\perp$ is exactly the space spanned by zero-mean data $\boldsymbol{x}_1 - \boldsymbol{\mu}, \boldsymbol{x}_2 - \boldsymbol{\mu}, \ldots, \boldsymbol{x}_N - \boldsymbol{\mu}$ with $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i$ being the global mean vector. $\mathbf{B}$ can be obtained by Gram-Schmidt ortho-normalization or standard PCA [2]. By replacing $\boldsymbol{\varphi}$ with its basis expansion (8) , we need to compute:

$$(\mathbf{B}^T\mathbf{S}_w\mathbf{B})\boldsymbol{\beta} = \mathbf{0} \quad . \tag{9}$$

The solution is *null space* of linear equations $\mathbf{B}^T\mathbf{S}_w\mathbf{B}$ spanned by $\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \ldots, \boldsymbol{\beta}^{(C)}$ with C being the number of classes.

By rewriting $\mathbf{S}_w$ as $\mathbf{X}_c\mathbf{X}_c^T$ with $\mathbf{X}_c$ representing the centralized data points corrected by the mean vector of each corresponding class $\mathbf{X}_c = \{\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \ldots, \bar{\boldsymbol{x}}_N\}$, $\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{\mu}^{(C_i)}$, $\forall i = 1, 2, \ldots, N$, with C_i indicating the class of sample i and $\boldsymbol{\mu}^{(k)}$ being the mean vector of class k, we are able to reformulate (9) as:

$$\mathbf{D}\mathbf{D}^T\boldsymbol{\beta} = \mathbf{0} \quad , \tag{10}$$

where $\mathbf{D} = \mathbf{B}^T\mathbf{X}_c$ consists of the dot products of basis vectors $\mathbf{B}$ and data points corrected by their mean vectors of corresponding classes, which suggests the kernelization.

It should be pointed out that when training data is incremented, both the orthogonal basis $\mathbf{B}$ and the within-class scatter $\mathbf{S}_w$ are altered and should be calculated all over again. Furthermore, the singular value decomposition of new $\mathbf{D}$ is very time-consuming.

By taking advantage of previously obtained information, the IKNDA algorithm proposed in this paper is able to update $\mathbf{D}$ and its null space in an efficient way. In the following section, both an incremental null space based LDA and its kernelization will be presented.

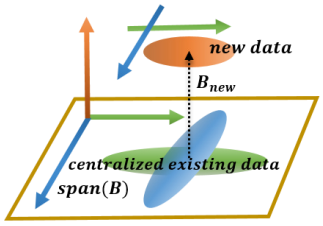


Figure 2: The new bases are perpendicular to the space spanned by centralized existing data samples.

4. Method Description

4.1. Incremental Null Space based LDA

For the purpose of incremental updating , we need to extract new information contained in newly-added data and update the model by an efficient scheme. Assume $\mathbf{X}$ is augmented by newly-added data set $\mathbf{Y}$. We denote $\mathbf{Z} = \{\mathbf{X}\ \mathbf{Y}\} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N, \boldsymbol{x}_{N+1}, \ldots, \boldsymbol{x}_{N+l}\}$ as the updated sample points where $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ represents the existing samples and $\mathbf{Y} = \{\boldsymbol{x}_{N+1}, \ldots, \boldsymbol{x}_{N+l}\}$ the newly-added points. $\mathbf{Z}_c$ is the centralized data matrix corrected by the mean vector of each class $\mathbf{Z}_c = \{\mathbf{X}_c, \mathbf{Y}_c\}$. Thus, we have $\mathbf{Z}_c = \mathbf{Z}\mathbf{\Pi}, \mathbf{\Pi} = \mathbf{I} - \mathbf{L}$, where $\mathbf{I}$ is an identity matrix and $\mathbf{L}$ is a block diagonal matrix with block sizes equal to the number of data points N_c in each class and the value $\frac{1}{N_c}$ at each position [2]. $\mathbf{H}_l = \mathbf{I}_l - \frac{1}{l}\mathbf{1}_l\mathbf{1}_l^T$ is a centralizing operator with $\mathbf{1}_l$ being an l-length all-one column vector. $\mathbf{Y}\mathbf{H}_l$ denotes the global mean-centralized new data $\{\boldsymbol{x}_{N+1} - \boldsymbol{\mu}_Y, \boldsymbol{x}_{N+2} - \boldsymbol{\mu}_Y, \ldots, \boldsymbol{x}_{N+l} - \boldsymbol{\mu}_Y\}$. The mean vectors of $\mathbf{X}$ and $\mathbf{Y}$ are, respectively, $\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y$. With these notations, the updated $\mathbf{D}$ can be formulated as follows:

$$\begin{pmatrix} \mathbf{B}^T\mathbf{X}_c & \mathbf{B}^T\mathbf{Y}_c \\ \mathbf{0} & \mathbf{B}_{new}^T\mathbf{Y}_c \end{pmatrix}, \tag{11}$$

where $\mathbf{B}_{new}$[1] is the new set of bases generated according to newly-added data points which can be extracted from $\bar{\mathbf{Y}} = \begin{pmatrix} \mathbf{Y}\mathbf{H}_l & \sqrt{\frac{Nl}{N+l}}(\boldsymbol{\mu}_X - \boldsymbol{\mu}_Y) \end{pmatrix}$. It consists of the centralized newly-added samples and a mean shift vector as well. The left bottom entry of (11) vanishes because the new bases are perpendicular to the space spanned by centralized existing data samples as illustrated by Figure 2, namely,

$$\mathbf{B}_{new}^T\mathbf{X}_c = \mathbf{0}. \tag{12}$$

4.2. Integrating with Kernel Trick

A fundamental assumption of Null Space method is the small sample size, i.e., $N < D$ [2], which is often unsatisfied in practical cases. To overcome this shortcoming, the kernel trick is frequently used, i.e., implicitly mapping the sample points into a high-dimensional space, and performing null space in this Reproducing Kernel Hilbert Space (RKHS). This procedure on the one hand effectively solves the problem mentioned above, on the other hand explores the nonlinear structure of the data.

In the following, we use $\mathbf{X}_\Phi$ to denote as the mapped sample points of existing samples $\mathbf{X}_\Phi = \{\Phi(\boldsymbol{x}_1), \Phi(\boldsymbol{x}_2), \ldots, \Phi(\boldsymbol{x}_N)\}$, and $\mathbf{Y}_\Phi$ to represent the mapped sample points of new samples $\mathbf{Y}_\Phi = \{\Phi(\boldsymbol{x}_{N+1}), \Phi(\boldsymbol{x}_{N+2}), \ldots, \Phi(\boldsymbol{x}_{N+l})\}$. Then, we have $\mathbf{Z}_\Phi = \{\mathbf{X}_\Phi\ \mathbf{Y}_\Phi\}$. Similarly, the centralized mapped data corrected by the corresponding mean vector of each class are denoted by $\mathbf{X}_\Phi^c$ and $\mathbf{Y}_\Phi^c$. $\boldsymbol{\mu}_{X_\Phi}$ and $\boldsymbol{\mu}_{Y_\Phi}$ are the mean vectors in high-dimensional space for $\mathbf{X}_\Phi$ and $\mathbf{Y}_\Phi$. With these notations, $\bar{\mathbf{Y}}_\Phi$ can be derived as follows:

$$\begin{aligned} \bar{\mathbf{Y}}_\Phi &= \begin{pmatrix} \mathbf{Y}_\Phi\mathbf{H}_l & \sqrt{\frac{Nl}{N+l}}(\boldsymbol{\mu}_{X_\Phi} - \boldsymbol{\mu}_{Y_\Phi}) \end{pmatrix} && (13) \\ &= (\mathbf{X}_\Phi\ \mathbf{Y}_\Phi)\begin{pmatrix} \mathbf{0}_N & \rho_1\mathbf{1}_N \\ \mathbf{H}_l & \rho_2\mathbf{1}_l \end{pmatrix} && (14) \\ &= (\mathbf{X}_\Phi\ \mathbf{Y}_\Phi)\begin{pmatrix} \mathbf{\Xi}_1 \\ \mathbf{\Xi}_2 \end{pmatrix} && (15) \\ &= \mathbf{Z}_\Phi\mathbf{\Xi}, && (16) \end{aligned}$$

where $\rho_1 = \sqrt{\frac{l}{N(N+l)}}, \rho_2 = -\sqrt{\frac{N}{l(N+l)}}, \mathbf{\Xi}_1 = \begin{pmatrix} \mathbf{0}_N & \rho_1\mathbf{1}_N \end{pmatrix}, \mathbf{\Xi}_2 = \begin{pmatrix} \mathbf{H}_l & \rho_2\mathbf{1}_l \end{pmatrix}$, $\mathbf{0}_N$ denotes a N-order square matrix of zeros, $\mathbf{1}_l$ is an l-length all-one column vector and similarly for $\mathbf{1}_N$.

Equation (11) still holds in the kernel induced feature space. Since the vectors in this space have infinite dimensions, $\mathbf{B}$ cannot be calculated explicitly. However, it can be reformulated as $\mathbf{B} = \mathbf{X}_\Phi\mathbf{V}_0$, where $\mathbf{V}_0 = \mathbf{H}\mathbf{Q}^r(\mathbf{\Delta}^r)^{-1/2}$ is derived by the rank-r eigen-decomposition of $\bar{\mathbf{K}}$ as $\mathbf{Q}^r\mathbf{\Delta}^r(\mathbf{Q}^r)^T$. Pairwise kernel dot products in high-dimensional space are collected in kernel matrix $\mathbf{K}$, namely, $\mathbf{K} = \mathbf{X}_\Phi^T\mathbf{X}_\Phi$. The kernel matrix is centralized as $\bar{\mathbf{K}} = \mathbf{H}\mathbf{K}\mathbf{H}$. $\mathbf{K}$ is guaranteed to be positive-definite by Reproducing Kernel Hilbert Theory. The first entry in (11) for the kernel induced feature space can therefore be formulated as:

$$\begin{aligned} \mathbf{B}^T\mathbf{X}_\Phi^c &= \mathbf{V}_0^T\mathbf{X}_\Phi^T\mathbf{X}_\Phi\mathbf{\Pi}_N && (17) \\ &= \mathbf{V}_0^T\mathbf{K}\mathbf{\Pi}_N && (18) \\ &= \mathbf{D}_0. && (19) \end{aligned}$$

The right upper element of (11) can also be rewritten in the same way:

$$\begin{aligned} \mathbf{B}^T\mathbf{Y}_\Phi^c &= \mathbf{V}_0^T\mathbf{X}_\Phi^T\mathbf{Y}_\Phi\mathbf{\Pi}_l && (20) \\ &= \mathbf{V}_0^T\mathbf{K}_1\mathbf{\Pi}_l && (21) \\ &= \mathbf{D}_1, && (22) \end{aligned}$$

[1] Note that $\mathbf{B}_{new}$ consists of new bases generated by samples corrected by global mean while $\mathbf{X}_c$ is the space spanned by samples corrected by the mean vector of each corresponding class.

where $\mathbf{K}_1$ consists of the pairwise dot products between existing data $\mathbf{X}_\Phi$ and newly-added data $\mathbf{Y}_\Phi$.

The remaining problem is the calculation of $\mathbf{B}_{new}$, the newly-introduced basis vectors that are perpendicular to the space spanned by the existing mean-corrected data. Since it cannot be calculated explicitly due to its infinite dimensionality, here we employ again the kernel trick as described in [6], and thus we can represent $\mathbf{B}_{new}$ as $\mathbf{Z}_\Phi \mathbf{V}_{new}$ taking advantage of the fact that $\mathbf{B}_{new}$ can be represented by linear combinations of updated data $\mathbf{Z}_\Phi$ with $\mathbf{V}_{new}$ being its coefficients. We firstly introduce some extra matrices:

$$\begin{aligned} \mathbf{\Gamma} &= \mathbf{B}^T \bar{\mathbf{Y}}_\Phi \\ \mathbf{\Psi} &= \bar{\mathbf{Y}}_\Phi - \mathbf{B}\mathbf{\Gamma}, \end{aligned} \tag{23}$$

where $\mathbf{\Gamma}$ contains the projection coefficients of $\bar{\mathbf{Y}}_\Phi$ onto the subspace spanned by $\mathbf{B}$, while $\mathbf{\Psi}$ contains new information in $\bar{\mathbf{Y}}_\Phi$, which are normal to the aforementioned subspace:

$$\mathbf{\Gamma} = \mathbf{B}^T \bar{\mathbf{Y}}_\Phi = \mathbf{V}_0^T {\mathbf{X}_\Phi}^T \mathbf{Z}_\Phi \mathbf{\Xi} \tag{24}$$

$$\mathbf{\Psi} = \bar{\mathbf{Y}}_\Phi - \mathbf{B}\mathbf{\Gamma} \tag{25}$$

$$= \begin{pmatrix} \mathbf{X}_\Phi & \mathbf{Y}_\Phi \end{pmatrix} \begin{pmatrix} \mathbf{\Xi}_1 - \mathbf{V}_0\mathbf{\Gamma} \\ \mathbf{\Xi}_2 \end{pmatrix} \tag{26}$$

$$= \mathbf{Z}_\Phi \mathbf{\Omega} \quad . \tag{27}$$

Instead of orthogonalizing $\mathbf{\Psi}$ directly to obtain $\mathbf{B}_{new}$, we perform eigen-decomposition on $\mathbf{\Psi}^T\mathbf{\Psi}$ to get a set of equivalent bases by collecting rank-r eigenvectors as in [6]:

$$\mathbf{B}_{new} = \mathbf{\Psi}\mathbf{Q}_\Psi \mathbf{\Delta}_\Psi^{-1/2} = \mathbf{Z}_\Phi \mathbf{V}_{new}, \tag{28}$$

where $\mathbf{\Psi}^T\mathbf{\Psi} = \mathbf{Q}_\Psi \mathbf{\Delta}_\Psi \mathbf{Q}_\Psi^T$ and $\mathbf{V}_{new} = \mathbf{\Omega}\mathbf{Q}_\Psi \mathbf{\Delta}_\Psi^{-1/2}$.

Therefore the last element in (11) can be rewritten as:

$$\mathbf{B}_{new}^T \mathbf{Y}_\Phi^c = \mathbf{V}_{new}^T \mathbf{Z}_\Phi^T \mathbf{Y}_\Phi \mathbf{\Pi}_l \tag{29}$$

$$= \mathbf{V}_{new}^T \mathbf{K}_2 \mathbf{\Pi}_l \tag{30}$$

$$= \mathbf{D}_2 \quad . \tag{31}$$

Note that $\mathbf{K}_2$ can be augmented from $\mathbf{K}_1$.

By using the kernel trick, equation (11) boils down to:

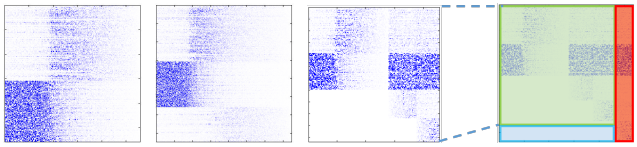
$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_0 & \mathbf{D}_1 \\ \mathbf{0} & \mathbf{D}_2 \end{pmatrix} \quad . \tag{32}$$

Having the solutions $\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \ldots, \boldsymbol{\beta}^{(C-1)}$, corresponding projections can be calculated by:

$$\boldsymbol{\varphi}^{(j)} = \mathbf{B}\boldsymbol{\beta}^{(j)} = \mathbf{Z}_\Phi [\mathbf{V}_0 \ \ \mathbf{V}_{new}] \boldsymbol{\beta}^{(j)}. \tag{33}$$

Having the updated coefficients $\mathbf{V} = [\mathbf{V}_0 \ \ \mathbf{V}_{new}]$, we have:

$$\boldsymbol{\varphi}^{(j)} = \mathbf{Z}_\Phi \mathbf{V} \boldsymbol{\beta}^{(j)} \quad . \tag{34}$$



(a) (b) (c)

Figure 3: An illustrative comparison of the batch KNDA with our IKNDA algorithm. (a): $\mathbf{V}^{k-1}$ in batch mode. (b): $\mathbf{V}^k$ in batch mode. Left part in (c): $\mathbf{V}^{k-1}$ in incremental mode. Right part in (c): $\mathbf{V}^k$ in incremental mode. As we can see, the batch method computes bases without taking advantage of previously computed matrix $\mathbf{V}^{k-1}$. While, our approach extracts new bases $\mathbf{V}_{new}$ from novel classes, marked in red square, then integrates with previously obtained information $\mathbf{V}^{k-1}$ (marked in green square).

The projected point of $\boldsymbol{x}$ is:

$$\begin{aligned} \boldsymbol{x}_j^* &= \boldsymbol{\varphi}^{(j)T} \Phi(\boldsymbol{x}) = \boldsymbol{\beta}^{(j)T} \mathbf{V}^T \mathbf{Z}_\Phi^T \Phi(\boldsymbol{x}) \\ &= \boldsymbol{\beta}^{(j)T} \mathbf{V}^T \boldsymbol{k}^* \quad \forall j = 1, 2, \ldots, C-1, \end{aligned} \tag{35}$$

where $\boldsymbol{k}^*$ stores the dot products between $\boldsymbol{x}$ and all the other existing data points in RKHS. During the updating process, only matrices $\mathbf{V}$ and $\boldsymbol{\beta}$ need to be updated. $\mathbf{V}$ can be obtained by integrating new information extracted from newly-added data and existing bases as illustrated in Figure 3. $\boldsymbol{\beta}$ is updated by invoking our proposed *incremental null space* method, more details of the algorithm will be discussed in the following section.

4.3. Incremental Null Space Updating

By (32) , it can be observed that the updated matrix $\mathbf{D}$ is augmented by $\mathbf{D}_1, \mathbf{D}_2$ from the existing matrix $\mathbf{D}_0$. Note that the null space of $\mathbf{D}\mathbf{D}^T$ is equivalent to the null space of $\mathbf{D}^T$.

The problem can then be tackled by the proposed incremental null space scheme. Let the null space of $\mathbf{D}^T$ be $\boldsymbol{\beta}$, then the product of $\mathbf{D}^T$ and $\boldsymbol{\beta}$ is computed as:

$$\mathbf{D}^T\boldsymbol{\beta} = \begin{pmatrix} \mathbf{D}_0^T & \mathbf{0} \\ \mathbf{D}_1^T & \mathbf{D}_2^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{pmatrix} = \mathbf{0} \quad . \tag{36}$$

From above we observe that $\boldsymbol{\beta}_1$ must lie in $\boldsymbol{\beta}_0$ which spans the null space of $\mathbf{D}_0^T$, since $\mathbf{D}_0^T\boldsymbol{\beta}_1 = \mathbf{0}$. Therefore $\boldsymbol{\beta}_1$ can be represented by linear combinations of $\boldsymbol{\beta}_0$:

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 \boldsymbol{\alpha} \quad . \tag{37}$$

Then, (36) can be rewritten as:

$$\begin{aligned} &\begin{pmatrix} \mathbf{D}_1^T\boldsymbol{\beta}_0 & \mathbf{D}_2^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta}_2 \end{pmatrix} = \mathbf{0} \\ &s.t. \quad \boldsymbol{\alpha}^T\boldsymbol{\alpha} + \boldsymbol{\beta}_2^T\boldsymbol{\beta}_2 = \mathbf{I} \quad , \end{aligned} \tag{38}$$

Figure 4: Representative images, two characters in six different font styles, of the FounderType-200 dataset.

which can be solved by employing linear equation solver or eigen-decomposition of matrix $\begin{pmatrix} \mathbf{D}_1^T\boldsymbol{\beta}_0 & \mathbf{D}_2 \end{pmatrix}$. Having $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_2$, the $\boldsymbol{\beta}$ is updated as $[(\boldsymbol{\beta}_0\boldsymbol{\alpha})^T \ \ (\boldsymbol{\beta}_2)^T]^T$.

The null space problem is much smaller scaled by our proposed scheme which has a complexity of $\boldsymbol{O}(l^2(c+b-1))$, where l is the incremental size, c is the number of classes and b is the number of new bases, compared with a complexity of $\boldsymbol{O}((N+l)^3)$ for the batch KNDA. Implementation details of our proposed algorithm are described as follows.

Algorithm : Incremental Kernel Null Space based DA

Initial Stage :

1: Centralize the kernel matrix : $\bar{\mathbf{K}} = \mathbf{HKH}$.
2: Obtain $\mathbf{V}^0 = \mathbf{HQ}^r(\boldsymbol{\Delta}^r)^{-1/2}$ by conducting eigen-decomposition of $\bar{\mathbf{K}} = \mathbf{Q}\boldsymbol{\Delta}\mathbf{Q}^T$.
3: Compute $\mathbf{D}_0 = (\mathbf{V}^0)^T\mathbf{K}\boldsymbol{\Pi}$.
4: Compute the null space $\boldsymbol{\beta}^0$ of matrix $\mathbf{D}_0^T$.

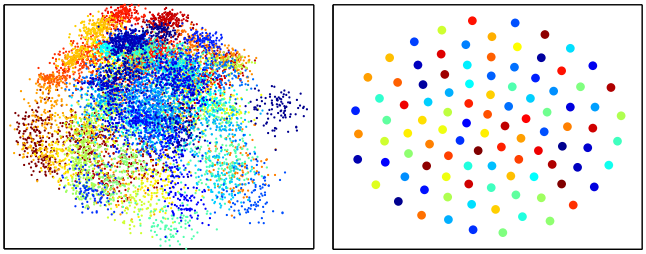
Updating Stage :

1: Compute $\mathbf{D}_1 = \mathbf{V}^{(k-1)^T}\mathbf{K}_1\boldsymbol{\Pi}_l$.
2: Compute new basis coefficient matrix $\mathbf{V}_{new}$.
3: Compute $\mathbf{D}_2 = \mathbf{V}_{new}^T\mathbf{K}_2\boldsymbol{\Pi}_l$.
4: Update $\boldsymbol{\beta}^k$ by the algorithm described in Section 3.3.
5: Update $\mathbf{V}^k$ by integrating new bases: $\mathbf{V}^k = [\mathbf{V}^{k-1} \ \ \mathbf{V}_{new}]$.
6: k ← k+1.

5. Time and Space Complexity

We can observe from the description presented above, main operations of our method are two parts: basic matrix multiplication and eigen-decomposition. As we know, the complexity of matrix product is typically $\boldsymbol{O}(mnp)$ for two matrices $\mathbf{A} \in \mathcal{R}^{m\times n}$ and $\mathbf{B} \in \mathcal{R}^{n\times p}$, while the complexity of eigen-decomposition is $\boldsymbol{O}(n^3)$ for a square matrix $\mathbf{C} \in \mathcal{R}^{n\times n}$.

Without losing generality, we assume that one novel class is injected in each iteration. By denoting the number of existing bases and new bases as $a = size(\mathbf{V}, 2)$ and $b = size(\mathbf{V}_{new}, 2)$, respectively, i.e., the column of matrix $\mathbf{V}$ and $\mathbf{V}_{new}$, we are able to analyze each step's complexity in **Updating Stage**. In step 1, only matrix multiplication is involved, which has a complexity of $\boldsymbol{O}(al(l+N))$. In step 2, an eigen-decomposition is performed whose complexity is $\boldsymbol{O}(l^3)$. In step 3, to obtain $\mathbf{D}_2$, a matrix product is conducted, costing $\boldsymbol{O}(bl(N+2l))$. Step 4 uses our proposed incremental null space algorithm and costs $\boldsymbol{O}(l^2(c+b-1))$, where c denotes the number of current classes. In general, l is relatively much smaller than N ($l \ll N$) and $b \ll a$. The complexity can be therefore reduced to $\boldsymbol{O}(l^3+alN)$, in which the time cost of implementing eigen-decomposition is $\boldsymbol{O}(l^3)$, and the total time spent to implement matrix product is $\boldsymbol{O}(alN)$. Compared with the complexity of $\boldsymbol{O}((l+N)^3)$ for eigen-decomposition of the batch KNDA method, the proposed approach is clearly more efficient.



(a) The CNN feature space. (b) Joint null space.

Figure 5: Joint null space of 100 classes in the FounderType-200 dataset. Each class in (a) is mapped to a single point in (b) (visualized by t-SNE).

	IKNDA	KNDA	SRKDA
time	$\boldsymbol{O}(l^3+alN)$	$\boldsymbol{O}((l+N)^3)$	$\boldsymbol{O}(N^2(l/2+c))$
space	$\boldsymbol{O}(Nl)$	$\boldsymbol{O}((l+N)^2)$	$\boldsymbol{O}(Nl)$

Table 2: Asymptotic complexity of IKNDA and the batch mode KNDA in terms of a, l, and N, where l is the incremental size, N is the number of existing samples, c is the number of classes and a is bounded by N.

6. Experiments

6.1. Experimental Setups

In this section, we carry out experiments to evaluate the performance of novelty detection methods on the following two publicly-available datasets: FounderType-200[2] and Caltech-256[3]. The FounderType-200 dataset we built consists of 200 different fonts produced by a company named FounderType with each font containing 6763 Chinese character images. Examples of this dataset are shown in Figure 4. Caltech-256 is composed of 256 categories with un-

[2] http://www.icst.pku.edu.cn/zlian/IKNLDA/
[3] http://www.vision.caltech.edu/Image_Datasets/Caltech256/

equal member sizes ranging from 80-800.

For FounderType-200, we randomly pick 100 fonts as the novel class, namely, these samples will not be used in training. While the remaining is split into training and test sets with equal size. Then we train a CNN network (i.e., Alexnet [13]) for feature extraction. Note that only the training set is utilized in the CNN training process. After training the CNN, all the samples are feed-forward and the output of the 7th fully connected layer is taken as features (4096 dims). For Caltech-256 dataset we pick 128 categorizes as the novel class and the rest is split into training and test sets in the similar way. For both datasets, the Radial Basis Function (RBF) is adopted for kernel construction.

To simulate the on-line updating process, we incrementally inject one class in every iteration. To perform novelty detection, we first map the test sample x to the null space as a single point x^*, and the corresponding novelty score is calculated as the smallest distance (Euclidean distance) between the point and all training class centers.

6.2. Results and Discussions

The learned CNN features of images in the FounderType-200 dataset are visualized in Figure 5a, from which we can see that large variance exists between samples within the same class. While in the joint null space (see Figure 5b), the variance vanishes for the same class, i.e., each class is mapped into one single point in this space. The probability of a given sample belonging to a known class can be simply measured by the distance between the sample and the mapped point of the class. This results in a more effective novelty score compared to the original feature space which explains the better performance (see Figure 6,7) of the proposed IKNDA compared to the method using a DNN classifier (i.e., Alexnet).

Figure 6 and Figure 7 plot the receiver operating characteristic (ROC) curves of different methods evaluated on FounderType-200 and Caltech-256 datasets, respectively. We compare our method with batch-KNDA [15], Spectral Regression KDA [4], DNN classifier and SVM in the one-vs-rest framework. The proposed IKNDA yields a ROC curve coincides exactly with the batch KNDA [15] which validates the effectiveness of our algorithm. For the FounderType-200 dataset, our method along with the batch mode, achieve the best result following by the Spectral Regression KDA [4] and DNN classifier. For the Caltech-256 dataset, our method achieves similar results as other methods compared here. Considering slighter difference between images in the same character content but different font styles, we believe that our method is more competitive for novelty detection tasks on large-scale fine-grain data.

We can also observe from the performance comparisons that our method along with other KDA approaches outperforms the DNN classifier in the original CNN feature space.

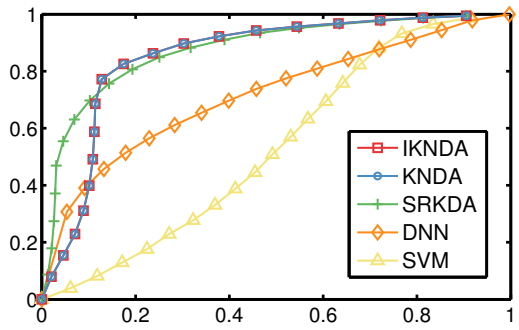


Figure 6: ROC curves of five novelty detection methods evaluated on the FounderType-200 dataset.

This suggests that the original CNN features is less capable of handling novelty detection tasks without proper transformations. Through our experiments, we can see that the null space based approach is well suited for this particular task.

As analyzed in section 5, the batch KNDA method shows a cubic growth against data size in terms of time complexity, while our proposed incremental algorithm is very efficient and mostly depends on the number of incremental size. Comparisons of novelty detection accuracy (measured by AUC values) are shown in Table 3 and 4. The AUC values and computational times of 5 different methods are listed in every iteration. In each iteration, we integrate 10 new classes into the existing model. It can be observed that our method achieves a comparable performance compared to the state of the art while significantly reduces the computational time. This validates the effectiveness and efficiency of the proposed IKNDA algorithm in applications of novelty detection for large-scale data. It should be pointed out that, as shown in Table 2, the time complexity of IKNDA is $\boldsymbol{O}(l^3+alN)$ and $\boldsymbol{O}(N^2(l/2+c)$ for SRKDA, where N and l denote the training size and incremental size, respectively. This means that the proposed IKNDA will also be much more efficient than SRKDA when there exist large numbers of training samples but less incremental data, which is commonly-seen in real applications.

Interestingly, we find that the AUC values of null space based methods are slightly lower than KDA in some iterations. This perhaps due to the fact that class variations are eliminated in the extracted null space which leads to overfitting. Yet, as we can see from our experimental results, the influence is slight.

7. Conclusion

This paper presented the incremental kernel null space based discriminant analysis (IKNDA) algorithm. We first briefly described the mathematical background of kernel null space based linear discriminant analysis. Then we de-

	AUC(%)					training time(s)			
#known	Ours	KNDA	SRKDA	SVM	DNN	Ours	KNDA	SRKDA	SVM
10	95.91	95.91	93.81	81.99	65.63	0.13	2.87	0.09	8.90
20	93.86	93.86	94.02	67.54	67.74	0.44	16.72	0.38	28.21
30	92.14	92.14	93.52	63.70	72.46	0.95	49.54	0.90	68.45
40	92.23	92.23	93.27	58.92	74.32	1.58	104.55	1.49	138.51
50	88.98	88.98	91.80	61.85	74.85	2.41	196.07	2.51	234.18
60	86.46	86.46	90.34	60.90	75.53	3.38	327.22	3.72	366.61
70	86.32	86.32	88.63	60.34	75.96	4.56	499.37	5.64	494.54
80	86.12	86.12	88.87	60.30	75.68	4.16	781.96	8.01	634.69
90	85.82	85.82	88.30	57.58	73.69	5.40	1047.56	12.22	838.94
100	85.59	85.55	87.86	53.46	71.85	9.25	1335.06	12.77	886.10

Table 3: AUC values and training times of five novelty detection methods evaluated on the FounderType-200 dataset.

	AUC(%)					training time(s)			
#known	Ours	KNDA	SRKDA	SVM	DNN	Ours	KNDA	SRKDA	SVM
10	87.90	87.90	88.39	80.14	77.52	0.03	0.56	0.02	2.43
20	86.19	86.19	88.03	83.23	80.33	0.09	2.85	0.10	10.20
30	83.55	83.55	84.93	82.87	77.53	0.20	7.61	0.21	22.65
40	83.37	83.37	85.11	82.41	79.39	0.34	15.76	0.37	43.67
50	82.75	82.75	84.50	81.78	79.00	0.54	31.05	0.55	74.76
60	81.37	81.37	83.28	80.73	78.25	0.82	48.55	0.99	116.77
70	80.24	80.24	82.36	80.21	77.95	1.10	73.93	1.23	170.44
80	79.50	79.33	81.31	79.69	78.46	1.46	109.51	1.91	236.06
90	78.87	78.87	80.34	77.96	78.40	1.87	160.48	2.85	310.80
100	79.07	79.07	80.87	78.75	79.38	2.25	233.38	2.95	403.29

Table 4: AUC values and training times of five novelty detection methods evaluated on the Caltech-256 dataset.

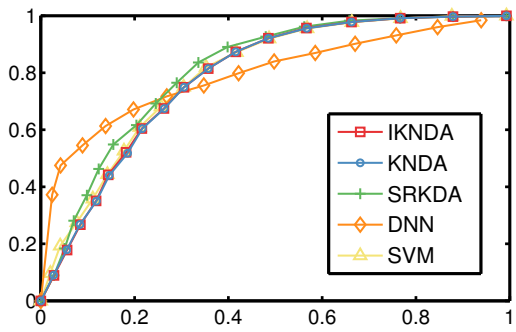


Figure 7: ROC curves of five novelty detection methods evaluated on the Caltech-256 dataset.

duced the incremental form of the standard null space based LDA. Finally, we proposed the IKNDA algorithm by incorporating the mathematical traits and an incremental null space method. Experimental results showed that the proposed algorithm obtains the performance that exactly coincides with the batch mode method while significantly reduces the time complexity in terms of the magnitude of order. The proposed method benefits from taking advantage of the existing model and computing only new bases brought by newly-added samples, then integrating them by an efficient updating scheme. Computational advantages of our method were proved theoretically as well as illustrated experimentally.

Our method is well suited for handling novelty detection tasks on large-scale datasets that might be successively injected and updated. Replacing batch computation with our IKNDA can accelerate these applications significantly. Furthermore, due to the incremental computations, our algorithm is far more scalable than the batch method.

Our future work will concentrate on the compression of samples as described in [6]. Note that even though the proposed algorithm sets us free from having to implementing the batch mode computation, we still have to store all samples in processing. Unnecessary storage can be reduced by leveraging the sample compression techniques.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant No.: 61672043, 61472015,

61672056 and 61402072), Beijing Natural Science Foundation (Grant No.: 4152022) and National Language Committee of China (Grant No.: ZDI135-9).

References

[1] E. Alexandre-Cortizo, M. Rosa-Zurera, and F. Lopez-Ferreras. Application of fisher linear discriminant analysis to speech/music classification. In *Computer as a Tool, 2005. EUROCON 2005. The International Conference on*, volume 2, pages 1666–1669. IEEE, 2005.

[2] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler. Kernel null space methods for novelty detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3374–3381. IEEE, 2013.

[3] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Computer VisionECCV 2002*, pages 707–720. Springer, 2002.

[4] D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 427–432. IEEE, 2007.

[5] T.-J. Chin, K. Schindler, and D. Suter. Incremental kernel svd for face recognition with image sets. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 461–466. IEEE, 2006.

[6] T.-J. Chin and D. Suter. Incremental kernel principal component analysis. *Image Processing, IEEE Transactions on*, 16(6):1662–1674, 2007.

[7] Y. A. Ghassabeh, F. Rudzicz, and H. A. Moghaddam. Fast incremental lda feature extraction. *Pattern Recognition*, 48(6):1999–2012, 2015.

[8] Y.-F. Guo, L. Wu, H. Lu, Z. Feng, and X. Xue. Null foley–sammon transform. *Pattern recognition*, 39(11):2248–2251, 2006.

[9] K. Hiraoka, K.-i. Hidai, M. Hamahira, H. Mizoguchi, T. Mishima, and S. Yoshizawa. Successive learning of linear discriminant analysis: Sanger-type algorithm. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, pages 664–667. IEEE, 2000.

[10] R. Huang, Q. Liu, H. Lu, and S. Ma. Solving the small sample size problem of lda. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 29–32. IEEE, 2002.

[11] E. A. K. James and S. Annadurai. Implementation of incremental linear discriminant analysis using singular value decomposition for face recognition. In *Advanced Computing, 2009. ICAC 2009. First International Conference on*, pages 172–175. IEEE, 2009.

[12] T.-K. Kim, K.-Y. K. Wong, B. Stenger, J. Kittler, and R. Cipolla. Incremental linear discriminant analysis using sufficient spanning set approximations. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[13] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural information processing systems*, pages 1097–1105, 2012.

[14] P. A. Lachenbruch. *Discriminant analysis*. Wiley Online Library, 1975.

[15] Y. Lin, G. Gu, H. Liu, and J. Shen. Kernel null foley-sammon transform. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering-Volume 01*, pages 981–984. IEEE Computer Society, 2008.

[16] W. Liu, Y. Wang, S. Z. Li, and T. Tan. Null space-based kernel fisher discriminant analysis for face recognition. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 369–374. IEEE, 2004.

[17] M. Markou and S. Singh. Novelty detection: a review part 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003.

[18] S. Marsland. Novelty detection in learning systems. *Neural Comp. Surveys*, 2003.

[19] C. Moulin, C. Largeron, C. Ducottet, M. Géry, and C. Barat. Fisher linear discriminant analysis for text-image combination in multimedia information retrieval. *Pattern Recognition*, 47(1):260–269, 2014.

[20] S. Pang, S. Ozawa, and N. Kasabov. Incremental linear discriminant analysis for classification of data streams. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(5):905–914, 2005.

[21] M. A. F. Pimentel, D. A. Clifton, C. Lei, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99(6):215–249, 2014.

[22] V. Roth. Kernel fisher discriminants for outlier detection. *Neural computation*, 18(4):942–960, 2006.

[23] A. Sharma and K. K. Paliwal. A new perspective to null linear discriminant analysis method and its fast implementation using random matrix multiplication with scatter matrices. *Pattern Recognition*, 45(6):2205–2213, 2012.

[24] N. Vaswani and R. Chellappa. Principal components null space analysis for image and video classification. *Image Processing, IEEE Transactions on*, 15(7):1816–1830, 2006.

[25] T. Xiong, J. Ye, Q. Li, R. Janardan, and V. Cherkassky. Efficient kernel discriminant analysis via qr decomposition. In *Advances in Neural Information Processing Systems*, pages 1529–1536, 2004.

[26] H. Yu and J. Yang. A direct lda algorithm for high-dimensional datawith application to face recognition. *Pattern recognition*, 34(10):2067–2070, 2001.

[27] H. Zhao and P. C. Yuen. Incremental linear discriminant analysis for face recognition. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(1):210–221, 2008.