Expert Systems With Applications 187 (2022) 115946

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Density-oriented linear discriminant analysis

Tahereh Bahraini [a,b], Seyed Mohammad Hosseini [c], Mahbubeh Ghasempour [b,d], Hadi Sadoghi Yazdi [b,d,*]

[a] Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[b] Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad, Mashhad, Iran
[c] Department of Electrical Engineering, Shiraz University, Shiraz, Iran
[d] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Keywords:
Linear discriminant analysis (LDA)
Outliers
Data density
Singularity problem
Adaboost
Robust LDA

ABSTRACT

The conventional Linear Discriminant Analysis (LDA) model has some challenges, such as sensitivity to the outlier, the singularity problem of the within-class scatter matrix, and Gaussian assumption of data within the same class. This paper proposes a robust LDA method that tries to solve the sensitivity to outliers and singularity problems. Specifically, we first use Bayesian risk to design the proposed method optimization problem. Then, the proposed Density-oriented LDA (DLDA) method used the data density as prior knowledge for robustness against outliers. The proposed method can classify non-linear and multi-mode distribution data sets. Furthermore, the proposed method can be employed for big data classification using the AdaBoost approach. Experimental results on synthetic and real data sets demonstrate the proposed DLDA method's superiority over other competing methods.

1. Introduction

The curse of dimensionality is one of the most critical challenges in utilizing the real high-dimensional multimedia data set and real applications, like video analysis (Zhao, Li, & Lu, 2019, 2020), object detection (Wang, Gao, & Yuan, 2017), and crowd analysis (Gao, Wang, & Yuan, 2019). Dimensionality reduction methods, e.g., principal component analysis (PCA) and linear discriminant analysis (LDA) (Rao, 1948) are widely used in signal processes and machine learning areas (Jia, Ma, & Gan, 2017; Peng, Qiao, Peng, & Wang, 2014) to solve the curse of dimensionality problem (Ye et al., 2016).

LDA method tries to learn a linear transformation matrix for mapping and maintaining the structure of original data. As a supervised feature reduction classifier, this method tries to obtain the optimal transformation matrix to maximize inter-class and minimize intra-class distances (Chen & Jin, 2012). The new extensions to this method are proposed as the new generation of classifiers in recent years which intend to solve the challenges of LDA, such as the outliers problem (Li et al., 2018; Zhou et al., 2015), Gaussian-mode-distribution (Nie, Wang, Wang, & Li, 2019; Nie, Wang, Wang, Wang, & Li, 2020), and the singularity of the within-class scatter matrix problems (Alimoglu & Alpaydin, 1997). For example, to reduce the sensitivity to the outliers, the existing literature presents some strategies, such as the use of neighborhood graph, L1-norm term, weighted classifier, regularization term, and change the LDA cost function (Zhong & Zhang, 2013). Also, the neighborhood-based method is applied to solve the single-mode-distribution problem (Duda, Hart, & Stork, 2001; Ye, 2007; Ye & Ji, 2010). The utilization of dimension reduction techniques such as PCA before applying the LDA classifier is a common approach to deal with the singularity problem. The other approach is based on the regularization term, known as regularization LDA or RLDA (Guo, Hastie, & Tibshirani, 2007). Chen et al. proposed a Null space of the within-class scatter matrix to solve this problem (Chen, Liao, Ko, Lin, & Yu, 2000). For real data with complex distribution, the average of different classes may change significantly due to the impact of outliers, and this seriously affects the LDA projection vector (Belhumeur, Hespanha, & Kriegman, 1997; Ye & Ji, 2010). In Yang, Yang, and Zhang (2008) and Ye and Yu (2005), the authors suggest the median instead of the mean value to overcome the outliers problem's effect. Allocate small weights to the outliers is a simple way to reduce their impact; however, the optimal value of these weights cannot be easily determined (Tang, Suganthan, Yao, & Qin, 2005). LODA algorithm decreased the outliers' effect by forming the neighborhood graph (Zhang & Chow, 2012). One

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

[*] Corresponding author at: Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.

E-mail addresses: bahraini.tahereh@mail.um.ac.ir (T. Bahraini), SeyedMohammad.Hosseini@shirazu.ac.ir (S.M. Hosseini), m.ghasempur@gmail.com (M. Ghasempour), h-sadoghi@um.ac.ir (H.S. Yazdi).

https://doi.org/10.1016/j.eswa.2021.115946
Received 12 March 2021; Received in revised form 30 August 2021; Accepted 19 September 2021
Available online 30 September 2021

reason for the LDA sensitivity against the outliers is the L2-norm term in its cost function. The use of L1-norm instead of this is suggested in Wang, Lu, Hu, and Zheng (2013).

Li et al. presented a transductive local Fisher discriminant analysis (TLFDA) algorithm (Li, Wang, Wang, Xue, & Wong, 2017) to use unlabeled data set in the learning process and improve the distinction capability. The geometric preserving LFDA (GeoPLFDA) algorithm assumed that data are on a non-linear manifold and the geometric structure interfered with the nearest neighborhood graph. By considering the sparsity in data space, different papers proposed sparse types of LDA. For example, the sparse LFDA (SLFDA) algorithm benefited from the features' sparseness to improve the distinction capability of LDA (Jia, Ruan, & Jin, 2016; Wang, Ruan, & An, 2016). The separability-oriented sub-class discriminant analysis (SSDA) is proposed in Wan, Wang, Guo, and Wei (2017) to obtain sub-classes more effectively and achieve higher class separation. The researchers proposed an effective iterative framework to solve an L1-norm minimization–maximization problem and developed an L1-norm distance measure based on LDA (L1-ELDA) (Ye et al., 2016). The authors showed that the L1-norm linear discriminant analysis (LDA-L1) algorithm is directly solved by a gradient ascending method, and non-greedy LDA-L1 (NLDA-L1) is a special case of improved LDA-L1 (ILDA-L1), which applies the same iterative procedure of ILDA-L1 (Ye, Zhao, Fu, & Gao, 2018). Yu et al. proposed the locality sensitive discriminant analysis (LSDA) with the group sparse representation for a hyperspectral image (HSI) classification (Yu, Gao, Li, Du, & Zhang, 2017). The authors solved some limitations of LDA by proposing a probabilistic MLDA method and developed an EM-type algorithm for parameter estimating (Cai & Huang, 2018). Other works done in this field are based on the Bhattacharyya error bound estimation and are of the weighted difference form of the within-class and the weighted pairwise between-class distances named L1BLDA and L2BLDA (Li, Shao, Wang, Deng, & Yang, 2019). A robust version of this method is RNNL2BLDA and uses reverse nearest neighbors (RNN) for multimodal data (Guo, Bai, Li, Shao, Ye, & Jiang, 2021). Also, robust sparse linear discriminant analysis (RSLDA) is proposed in Wen et al. (2018). RSLDA overcomes the challenges of LDA using an adaptive selection of the most discriminative features via the l2,1 norm, orthogonal and sparse matrices.

This paper is organized as follows: In Section 2 the related works are reviewed. The proposed density-oriented LDA method (DLDA) is introduced in Section 3. Also, in this section, four matters for the proposed DLDA method are investigated: the effect of outliers in the performance of the proposed DLDA method in sub-Section 3.1, the robust version of DLDA method in sub-Section 3.2, the time complexity in Section 3.3, two designed strategies to improve the performance of the proposed DLDA method for classifying big data sets – called ABDLDA and ABWDLDA – in sub-Section 3.4. In Section 4, the experimental results are described. Finally, the conclusion is summarized in Section 5.

2. Related works

2.1. Linear Regression (LR) method

Let us suppose one data set with two classes that consists of n samples and can be represented by $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in R^d$ is data sample and $y_i \epsilon \{-1, 1\}$ is label of this sample. The ith output of LR method – which is equivalent to two-class LDA – has the estimated label $f(x_i) = x_i^T w + b$, where $w \in R^d$ and b are the weight vector and bias, respectively. The general approach to estimate optimal values of w and b is based on the minimization of the least square cost function as:

$$\begin{aligned} \{w^*, b^*\} &= \arg\min_{w,b} L(w, b) = \arg\min_{w,b} \frac{1}{2} \|X^T w + 1b - Y\|^2 \\ &= \arg\min_{w,b} \frac{1}{2} \sum_{i=1}^{n} \|f(x_i) - y_i\|^2. \end{aligned} \tag{1}$$

where $X = [x_1, x_2, \ldots, x_n]^T$, and $Y = [y_1, y_2, \ldots, y_n]^T$ are data matrix and the label vector, respectively. Also, **1** is equal to $\mathbf{1} = [1, 1, \ldots, 1]^T$,. $\{x_i\}_{i=1}^{n}$, and $\{y_i\}_{i=1}^{n}$ are centralized data with zero means. Let us form label vector as $y_i \in \left\{ \frac{-2n}{n_2}, \frac{2n}{n_1} \right\}$, where n_1 and n_2 represent the number of samples in each class. Assume that $b = 0$, so Eq. (1) can be rewritten as:

$$\{w^*\} = \arg\min_{w} L(w) = \arg\min_{w} \frac{1}{2} \|X^T w - Y\|^2. \tag{2}$$

The optimal weight for this optimization problem is obtained as $w^* = (XX^T)^+ XY$, where $(.)^+$ denotes the pseudo-reverse operator. Since X is the zero mean matrix, we have $XX^T = nS_t$, where S_t is called total scatter matrix and defined as $S_t = \frac{1}{n} \sum_{i=1}^{n} (x_i - c)(x_i - c)^\tau$. Therefore, we have $XY = \frac{2n_1 n_2}{n^2} (c^{(1)} - c^{(2)})$, where $c^{(1)}$ and $c^{(2)}$ represent the mean of classes C_1 and C_2, respectively. Then, we have $w^* = \frac{2n_1 n_2}{n^2} S_t^+ (c^{(1)} - c^{(2)}) = \frac{2n_1 n_2}{n^2} G^F$, where G^F is the optimal solution for two-class LDA.

2.2. Multivariate Linear Regression (MLR) method

Consider a multi-class data set with n samples $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in R^d$, $y_i \in \{1, 2, \ldots, k\}$, and $k > 2$ indicates the number of classes. Let n_i denotes the samples number of ith class, and each element of indicator matrix $\mathcal{Y} \in R^{n \times k}$ is equal to $\mathcal{Y}(ij) = \begin{cases} 1 & y_i = j, \\ -1/(k-1) & \text{o.w.} \end{cases}$. Suppose $\tilde{X} = [\widetilde{x}_1, \ldots, \widetilde{x}_n] \in R^{d \times n}$, and $\widetilde{\mathcal{Y}} = [\widetilde{\mathcal{Y}_1}, \ldots \widetilde{\mathcal{Y}_k}] \in R^{n \times k}$ be the centralized matrices equivalent to data matrix X, and indicator matrix $\mathcal{Y}$, where the jth column of $\widetilde{\mathcal{Y}}$ is $\widetilde{\mathcal{Y}_j} = [\mathcal{Y}_{(1j)}, \ldots, \mathcal{Y}_{(ij)}, \ldots, \mathcal{Y}_{(nj)}]^T$. The elements of these two matrices are $\widetilde{x}_i = x_i - \frac{1}{n} \sum_{i=1}^{n} x_i$ and $\widetilde{\mathcal{Y}}_{ij} = \mathcal{Y}_{ij} - \frac{1}{n} \sum_{i=1}^{n} \mathcal{Y}_{ij}$. In the MLR method, k discriminant functions are defined for ith sample as $f(x_i) = (f_1(x_i), f_2(x_i), \ldots, f_k(x_i))$. also, the weight matrix $W = [w_1, w_2, \ldots, w_k | \{w_j\}_{j=1}^{k} \in R^d]$ is defined based on the k linear models $f_j(x) = x^T w_j$ and calculated by minimizing the sum of squares error function as:

$$\begin{aligned} W^* &= \arg\min_{W} L(W) = \arg\min_{W} \frac{1}{2} \|\widetilde{X}^T W - \widetilde{\mathcal{Y}}\|^2 \\ &= \arg\min_{w} \frac{1}{2} \sum_{j=1}^{k} \sum_{i=1}^{n} \|f_j(\widetilde{x}_i) - \widetilde{\mathcal{Y}}_{ij}\|^2 \end{aligned} \tag{3}$$

The optimal value of this problem is $W^* = (\widetilde{X}\widetilde{X}^T)^+ \widetilde{X}\widetilde{\mathcal{Y}}$, which is also known as Weiner's weight.

3. The proposed DLDA method formulation

In the proposed DLDA method, we try to solve the challenges of LDA, such as the effect of outliers and the singularity problem of the scatter matrix, by using the prior knowledge of data distribution. Let us assume $X = [x_1, \ldots, x_n]^T$ and $Y = [y_1, \ldots, y_n]^T$ are the data samples and label matrix, where $x_i \in R^d$ and $y_i \in R^l$ are the ith sample and its label. Our goal is to map each sample x_i from a d-dimensional space to y_i in the l-dimensional space. This projection is done using matrix $W \in R^{d \times l}$ and function f, where $l < d$ and $y_i = f(x_i) = W^T x_i$, $\forall i = 1, \ldots, n$. Let we assume $y_i = W^T x_i + \mathcal{N}_i$ is model of signal, where $\mathcal{N}_i$ is additive white Gaussian noise with pdf $\mathcal{N}_i \sim N(0, \sigma^2_{\mathcal{N}_i})$. Also, W is Gaussian random variable with pdf $W \sim N(0, \sigma^2_W)$. We used the Bayesian Risk $R_{Bayes}(W, W^*)$ for each sample x_i to propose the optimization problem and obtain the optimal value of W:

$$W^* = \arg\min_{W} R_{Bayes}(W, W^*) = \arg\min_{W} E\{L(W, W^*)\}, \tag{4}$$

where E and L denote expectation operator and loss function $L(W, W^*) = 1 - \delta(W, W^*)$, in which $\delta(W, W^*)$ is Dirac delta function. Eq. (4) continues as follows:

$$W^* = \arg\min_{W} \left\{ \int_W \int_y L(W, W^*) f_{W,y}(W, y) dW dy \right\} \tag{5}$$

where, $f_{W,y}(W,y)$ is jointly pdf of two variables W and y, and is equal to $f_{W,y}(W,y) = f_{W|y}(W|y)f_W(W)$. Let us assume that $f_W(W)$ is constant. So, Eq. (5) is simplified as:

$$\begin{aligned} W^* &= \arg\min_W \left\{ \int_W L(W, W^*) f_{W|y}(W|y) dW \right\} \\ &= \arg\min_W \{1 - f_{W|y}(W|y)\} = \arg\max_W f_{W|y}(W|y) \end{aligned} \tag{6}$$

Then, using pdf and conditional pdf of two variable W and y, we have:

$$\begin{aligned} W^* &= \arg\max_W f_{y|W}(y|W) f_W(W) \\ &= \arg\max_W \left\{ \frac{1}{\sqrt{2\pi\sigma^2_{\mathcal{N}_i}}} \exp\left(-\frac{\|y_i - W^T x_i\|^2}{2\sigma^2_{\mathcal{N}_i}}\right) \times \frac{1}{\sqrt{2\pi\sigma^2_W}} \exp\left(-\frac{\|W\|^2}{2\sigma^2_W}\right) \right\} \end{aligned} \tag{7}$$

By simplifying this equation, we reach the following optimization problem for each sample $(x_i, y_i), \forall i \in \{1, \dots, n\}$:

$$\begin{aligned} W^* &= \arg\min_W \left\{ \frac{\|y_1 - W^T x_1\|^2}{2\sigma^2_{\mathcal{N}_1}} \times \frac{\|W\|^2}{2\sigma^2_W} \right\} \\ &\vdots \\ W^* &= \arg\min_W \left\{ \frac{\|y_n - W^T x_n\|^2}{2\sigma^2_{\mathcal{N}_n}} \times \frac{\|W\|^2}{2\sigma^2_W} \right\} \end{aligned} \tag{8}$$

Therefore, the general optimization problem for all samples can be rewritten as follows:

$$W^* = \arg\min_W \left\{ \frac{1}{n} \sum_{i=1}^{n} p(x_i) \|W^T x_i - y_i\|^2 + \alpha \|W\|_k^2 \right\} \tag{9}$$

where $\alpha = \frac{1}{2\sigma^2_W}$ is regularization parameter, and $p(x_i)$ is the probability density of ith data sample x_i that is usually unknown and must be estimated. In the training phase, each sample has a unique impact on finding the optimal weight W^*; some of them have a positive effect, and some have a destructive effect. Among these, positive impact samples are the main part of training data and almost without noise. These samples have a probability density close to each other and are more probable to occur. And the other part of training samples harms finding optimal weight named noisy samples or outliers, which have less probability of occurrence. We use a fuzzy degree to determine the importance of each sample in the interval $0 < p(x_i) \leq 1$. If x_i is one of the main training samples (no outlier), the probability of occurrence of this sample is high. The density $p(x_i)$ in (9) is increased; thus, x_i affects more than outliers on the minimization problem, which is solved more accurately. This leads to overcome the outlier sensitivity and singularity problems.

In this paper, the Parzen windows method is employed to estimate $p(x_i)$, which is calculated as follows:

$$p(x_i) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi\left(\frac{x_i - x_j}{h}\right). \tag{10}$$

where ϕ is the Parzen window. Let us define the probability density matrix $P(X)$ as a $n \times n$ diagonal matrix which contains $p(x_i)$:

$$P(X) = \begin{bmatrix} p(x_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & p(x_n) \end{bmatrix} \tag{11}$$

So, Eq. (9) is rewritten as:

$$W^* = \arg\min_W \left\{ \frac{1}{n} (X^T W - Y)^T P(X) (X^T W - Y) + \alpha W^T W \right\} \tag{12}$$

By optimizing this problem, the optimal value of our proposed DLDA method is obtained using $\frac{\partial J(W)}{\partial W} = 0$ as follow:

$$\begin{aligned} &2XP(X)X^T W - XP(X)Y - (Y^T P(X) X^T)^T + 2\alpha W = 0, \\ &W^* = (XP(X)X^T + \alpha I)^{-1} XP(X)Y. \end{aligned} \tag{13}$$

By substituting $y_i \in \left\{ \frac{-2n}{n_2}, \frac{2n}{n_1} \right\}$ and $x_i p(x_i)$ with x'_i into (13), we get:

$$XP(X)Y = -\frac{2n_1 n_2}{n} \sum_{\langle C_1 \rangle} \frac{x'_i}{n_1} + \frac{2n_1 n_2}{n} \sum_{\langle C_2 \rangle} \frac{x'_i}{n_2} = \frac{2n_1 n_2}{n}(m'_2 - m'_1). \tag{14}$$

where $m'_1 = \sum_{\langle C_1 \rangle} \frac{x_i p(x_i)}{n_1}$ and $m'_2 = \sum_{\langle C_2 \rangle} \frac{x_i p(x_i)}{n_2}$ are the mean of classes C_1 and C_2, respectively. Suppose that the mean of data is zero; therefore, the total scatter matrix is obtained as:

$$S'_t = \frac{1}{n} \sum_{i=1}^{n} x_i p(x_i) x_i^T. \tag{15}$$

according to (13), (14), and (15), we have the projection matrix W^* for the proposed DLDA method as:

$$W^* = \frac{2n_1 n_2}{n^2} (S'_t + \alpha I)^{-1} (m'_2 - m'_1) \tag{16}$$

Algorithm I: Proposed DLDA Algorithm

```
Input: The training data samples: $X = [x_1, \dots, x_n]^T$ where $x_i \in R^d$, label: $Y = [y_1, \dots, y_n]^T$ where $y_i \in R^l$.
Output: Transformation matrix: $W^* \in R^{d \times l}$.
Initialization: Set initial values to $\alpha$, $h$.
for $i = 1 : n$
    Calculate $p(x_i)$ as probability density of $i$th sample $x_i$ using Eq. (10)
end for
Calculate $m'_1 = \sum_{\langle C_1 \rangle} \frac{x_i p(x_i)}{n_1}$ and $m'_2 = \sum_{\langle C_2 \rangle} \frac{x_i p(x_i)}{n_2}$ as the mean value of two class $C_1$ and $C_2$
Calculate $S'_t$ using Eq. (15)
Calculate $W^*$ using Eq. (16)
```

Also, we used the Binary Hierarchical Tree Balanced Branches (BHT-BB) strategy to expand the binary DLDA classifier into the multi-class model (called MDLDA). The proposed DLDA method is summarized in algorithm I.

3.1. Effect of outliers

If we assume that x_i is an outlier sample, its probability density is expressed as $p(x_i) \leq \epsilon \frac{1}{nh^d}$, where ϵ is a positive small value. Let us assume that this outlier belongs to the second class C_2; therefore, mean of this class (i.e., m''_2) is updated as follows:

$$m''_2 = m'_2 + \frac{1}{n+1}\left(x\frac{\epsilon}{nh^d} - m'_2\right). \tag{17}$$

Also, this outlier affects the total scatter matrix S'_t. So, S''_t is updated as follows:

$$S''_t = S'_t + x\frac{\epsilon}{nh^d}x^T. \tag{18}$$

By using Eqs. (17) and (18), new projection matrix $W^{*\prime}$ will be obtained as:

$$W^{*\prime} = \frac{2n_1 n_2}{n^2}(S''_t + \alpha I)^{-1}(m''_2 - m'_1) = \frac{2n_1 n_2}{n^2}\left(S'_t + x\frac{\epsilon}{nh^d}x^T + \alpha I\right)^{-1}(m''_2 - m'_1). \tag{19}$$

Lemma 1 (*Woodbury Matrix Identity*). *If matrix A can be written in this form $A = B^{-1} + CD^{-1}C^T$, its inverse is equal to $A^{-1} = B - BC(D + C^T BC)^{-1} C^T B$.*

In Eq. (19), we assumed that $A = \tilde{S}_t + x\frac{\epsilon}{nh^d}x^T$ where $\tilde{S}_t = S'_t + \alpha I$. Using Lemma 1, the inverse of term $(S'_t + x\frac{\epsilon}{nh^d}x^T + \alpha I)$ is obtained as: $A^{-1} = \tilde{S}_t^{-1} - \tilde{S}_t^{-1}x(\frac{nh^d}{\epsilon} + x^T\tilde{S}_t^{-1}x)^{-1}x^T\tilde{S}_t^{-1}$. So, the projection matrix can be reformulated as:

$$W^{*\prime} = \frac{2n_1n_2}{n^2}\tilde{S}_t^{-1}(m''_2 - m'_1) - \frac{2n_1n_2}{n^2}\tilde{S}_t^{-1}x(\frac{nh^d}{\epsilon} + x^T\tilde{S}_t^{-1}x)^{-1}x^T\tilde{S}_t^{-1}(m''_2 - m'_1). \tag{20}$$

We will prove that the outliers have no significant effect on the mean value. Hence, we have $W^{*\prime}$ as:

$$W^{*\prime} = W(I - M). \tag{21}$$

where $W = \frac{2n_1n_2}{n^2}\tilde{S}_t^{-1}(m''_2 - m'_1)$ and $M = \tilde{S}_t^{-1}x(\frac{nh^d}{\epsilon} + x^T\tilde{S}_t^{-1}x)^{-1}x^T$. If we show that $I - M \approx I$, then $W^{*\prime} \approx W^*$. Consequently, the new projection matrix $W^{*\prime}$ will be almost equal to the previous projection matrix W, and we need to estimate the value of $\tilde{S}_t^{-1}$. Therefore, We can expressed $\tilde{S}_t x = \alpha x$, and then $\tilde{S}_t^{-1}\tilde{S}_t x = \tilde{S}_t^{-1}\alpha x \longrightarrow \frac{1}{\alpha}x = \tilde{S}_t^{-1}x$. So, we have:

$$x^T\tilde{S}_t^{-1}x = x^T\frac{1}{\alpha}x \longrightarrow \frac{1}{\alpha}x^Tx = \frac{1}{\alpha}\|x\|^2. \tag{22}$$

where $\|x\|^2$ is distance of each point from samples' center, (because $\{x_i\}$ and $\{y_i\}$ are centered, i.e., each point is deducted from the total mean of data). Also, α is the eigenvalue of data and related to data elongation. α increases for the outlier sample and decreases for the main samples. For the data axis with a long stretch, $\frac{\|x\|^2}{\alpha}$ has a small value; in other words, the outliers will have less effect. But it will have more effect on the direction of the eigenvectors, which have small values of α. Therefore, there are three modes for M:

Mode 1: If $\frac{nh^d}{\epsilon} > \frac{1}{\alpha}\|x\|^2$, $\frac{1}{\alpha}\|x\|^2$ can be ignored, and we have $M = \tilde{S}_t^{-1}x\frac{\epsilon}{nh^d}x^T = \frac{\epsilon}{nh^d}\tilde{S}_t^{-1}xx^T$. If we assume that xx^T is an estimate of $\tilde{S}_t$, so $\tilde{S}_t^{-1}xx^T \approx I$, and therefore $M \approx \frac{\epsilon}{nh^d}$. Using this expression for M, we can say that this has a small value, and with increasing number of samples n and choosing acceptable h or increasing the data dimension, its value becomes smaller. We obtained $I - M \approx I$, so $W^{*\prime} \approx W^*$ is established.

Mode 2: If $\frac{nh^d}{\epsilon} = \frac{1}{\alpha}\|x\|^2$, then we have $M = \tilde{S}_t^{-1}x(\frac{nh^d}{\epsilon} + \frac{nh^d}{\epsilon})^{-1}x^T = \tilde{S}_t^{-1}x\frac{\epsilon}{2nh^d}x^T$. It looks like the first mode, so we have the same result $W^{*\prime} \approx W^*$.

Mode 3: $\frac{nh^d}{\epsilon} < \frac{1}{\alpha}\|x\|^2$ does not occur for outliers. Because the outliers have a high elongation in their direction, and they have a large value of eigenvalues α in the direction of the eigenvectors of $\tilde{S}_t$. So, the value of $\frac{1}{\alpha}\|x\|^2$ is small for outliers. However, the elongation is low for the non-outlier samples, and the value of $\frac{1}{\alpha}\|x\|^2$ is noteworthy. So, it is proved that the outliers do not make a significant change in the projection axis of the proposed DLDA method.

3.2. Robust DLDA method

In Eq. (9), a square loss function which is useful for Gaussian noise exposure is appeared. However, the correntropy loss function $L(e_i) = 1 - \exp(-\sigma_i^2\|f(x_i) - y_i\|^2)$ is proposed for robustness against outliers and to reduce non-Gaussian noises (Liu, Pokharel, & Principe, 2007). Where e_i and $\sigma_i \in [0, 1]$ are the ith error of ith sample and the scaling parameter, respectively. So, optimization problem (9) can be rewritten:

$$\begin{aligned}\{W^*, \sigma_i\} &= \arg\min_{W,\sigma_i}\{\frac{1}{n}\sum_{i=1}^{n}p(x_i)(1 - \exp(-\sigma_i^2\|W^Tx_i - y_i\|^2)) + \alpha\|W\|_k^2\}\\ s.t. &\quad \sum_{i=1}^{n}\sigma_i = \eta\end{aligned} \tag{23}$$

where this is equivalent to

$$\begin{aligned}\{W^*, \sigma_i\} &= \arg\max_{W,\sigma_i}\{\frac{1}{n}\sum_{i=1}^{n}p(x_i)\exp(-\sigma_i^2\|W^Tx_i - y_i\|^2) - \alpha\|W\|_k^2\}\\ s.t. &\quad \sum_{i=1}^{n}\sigma_i = \eta\end{aligned} \tag{24}$$

We use the half-quadratic solution method (Boyd, Boyd, & Vandenberghe, 2004) to solve this optimization problem. According to the conjugate function theory of this method, we have:

$$\exp(-\sigma_i^2\|W^Tx_i - y_i\|^2) = \sup_{\mathfrak{p}_i<0}(\sigma_i^2\|W^Tx_i - y_i\|^2\mathfrak{p}_i - \phi(\mathfrak{p}_i)) \tag{25}$$

where $\phi(\mathfrak{p}_i)$ is a convex function and is equal to $\phi(\mathfrak{p}_i) = -\mathfrak{p}_i\log(-\mathfrak{p}_i) + \mathfrak{p}_i$. Therefore, Eq. (24) can be rewritten as:

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}p(x_i)\sup_{\mathfrak{p}_i<0}(\sigma_i^2\|W^Tx_i - y_i\|^2\mathfrak{p}_i - \phi(\mathfrak{p}_i)) - \alpha\|W\|_k^2\\ &= \sup_{\mathfrak{p}_i<0}\{\frac{1}{n}\sum_{i=1}^{n}p(x_i)(\sigma_i^2\|W^Tx_i - y_i\|^2\mathfrak{p}_i - \phi(\mathfrak{p}_i))\} - \alpha\|W\|_k^2\\ &= \sup_{\mathfrak{p}_i<0}\{\frac{1}{n}\sum_{i=1}^{n}p(x_i)(\sigma_i^2\|W^Tx_i - y_i\|^2\mathfrak{p}_i - \phi(\mathfrak{p}_i)) - \alpha\|W\|_k^2\}\end{aligned} \tag{26}$$

In (26), the second equation establishes because $\sigma_i^2\|W^Tx_i - y_i\|^2\mathfrak{p}_i - \phi(\mathfrak{p}_i)$; $i \in \{1, \dots, n\}$ are independent functions in terms of $\mathfrak{p}_i$. In addition, the third equation establishes since $-\alpha\|W\|_k^2$ is constant with respect to $\mathfrak{p}_i$. So, using (24) and (26) we have:

$$\begin{aligned}\{W^*, \sigma_i, \mathfrak{p}_i^*\} &= \arg\max_{W,\sigma_i,\mathfrak{p}_i<0}\{\frac{1}{n}\sum_{i=1}^{n}p(x_i)(\sigma_i^2\|W^Tx_i - y_i\|^2\mathfrak{p}_i - \phi(\mathfrak{p}_i)) - \alpha\|W\|_k^2\}\\ s.t. &\quad \sum_{i=1}^{n}\sigma_i = \eta\end{aligned} \tag{27}$$

We use the alternating approach to find the optimal values of three variables W, σ_i and $\mathfrak{p}_i$ separately, while the others are assumed to be fixed. Therefore, we have three following sub-problems:

(1) W-sub-problem: We can get the optimal value of W by solving the following problem:

$$W^* = \arg\max_{W}\{\frac{1}{n}\sum_{i=1}^{n}p(x_i)\sigma_i^2\|W^Tx_i - y_i\|^2\mathfrak{p}_i - \alpha\|W\|_k^2\} \tag{28}$$

By defining $q_i = -\mathfrak{p}_i = \exp(\sigma_i^2\|W^Tx_i - y_i\|^2)$ for $i \in \{1, \dots, n\}$, and also $p'(x_i) = p(x_i)\sigma_i^2q_i$, we will have:

$$W^* = \arg\min_{W}\{\frac{1}{n}\sum_{i=1}^{n}p'(x_i)\|W^Tx_i - y_i\|^2 - \alpha\|W\|_k^2\} \tag{29}$$

Let us rewrite the matrix form of $p'(x_i)$ as a $n\times n$ diagonal matrix $P'(X)$. Therefore, the matrix form of the problem in (28) is obtained as follows:

$$W^* = \arg\min_{W}\left\{\frac{1}{n}(X^TW - Y)^TP'(X)(X^TW - Y) + \alpha W^TW\right\} \tag{30}$$

We can solve this equation as:

$$W^* = (XP'(X)X^T + \alpha I)^{-1}XP'(X)Y \tag{31}$$

This solution can be displayed by simplification and placement as follows:

$$W^* = \frac{2n_1n_2}{n^2}(Z_t + \alpha I)^{-1}(a_2 - a_1) \tag{32}$$

where $a_1 = \sum_{\langle C_1\rangle}\frac{x_ip'(x_i)}{n_1}$, $a_2 = \sum_{\langle C_2\rangle}\frac{x_ip'(x_i)}{n_2}$ and $Z_t = \frac{1}{n}\sum_{i=1}^{n}x_ip'(x_i)x_i^T$.

(2) $\mathfrak{p}_i < 0$-sub-problem: In this step, p_i is updated and other variables are assumed to be fixed, so to optimize p_i, the problem (27) becomes:

$$\mathfrak{p}_i^* = \arg\max_{\mathfrak{p}_i<0}\{\frac{1}{n}\sum_{i=1}^{n}p(x_i)(\sigma_i^2\|W^Tx_i - y_i\|^2\mathfrak{p}_i - \phi(\mathfrak{p}_i))\} \tag{33}$$

The analytic solution of Eq. (33) is

$$\mathrm{p}_i^* = -\exp(-\sigma_i^2 \| W^T x_i - y_i \|^2) \tag{34}$$

(3) σ_i-sub-problem: The optimization problem with respect to σ_i is:

$$\begin{aligned} \sigma_i^* &= \arg\min_{\sigma_i} \{ \frac{1}{n} \sum_{i=1}^{n} p(x_i) \sigma_i^2 \| W^T x_i - y_i \|^2 q_i \} \\ s.t. \quad & \sum_{i=1}^{n} \sigma_i = \eta \end{aligned} \tag{35}$$

where η is a manual parameter and is defined by the user. By adding $-\lambda(\sum_{i=1}^{n} \sigma_i - \eta)$ as the penalty term in the Lagrangian form and solving the obtained problem, we have

$$\sigma_i^* = \frac{\lambda}{\frac{1}{n} p(x_i) \| W^T x_i - y_i \|^2 q_i} \tag{36}$$

3.3. Time complexity

The time complexity of the training phase for two-class LDA classifier is equivalent to $O(d^2 n)$, where n is the number of training samples with d-dimensional (Duda et al., 2001). For the proposed DLDA method, the time complexity of the training phase is equal to $O(dn^2 + d^2 n + n^2)$, part of which (equal to $O(n^2)$) is related to the calculation of probability density $p(x_i)$ for training samples. Therefore, the proposed DLDA classifier has more time complexity than the LDA due to calculating $p(x_i)$; however, it is not very impressive for the small data sets. Also, to solve this problem for big data sets, the probability density of data samples is calculated locally. There is no need to calculate all training samples' probability density, which is discussed in the following sections. For the proposed MDLDA method, if the number of classes is c, the time complexity of this method is equal to $O(c^2 \log c(dn^2 + d^2 n + n^2))$.

3.4. Proposed DLDA method for big data set

A combination of Adaboost and weighted Adaboost methods (Freund & Schapire, 1997; Freund, Schapire, & Abe, 1999; Schapire & Singer, 1999) with the proposed DLDA method is suggested to classify the nonlinear big data sets.

The Adaboost method is a weighted combination form of T basis classifiers. For the ith data sample x_i, it is defined as follows:

$$F(x_i) = \sum_{t=1}^{T} w_t f_t(x_i). \tag{37}$$

where w_t is the weight of the classifier $f_t(x_i)$. In this method, data is broken into several subsets, and DLDA classifier applies to each of these. The overall process of Adaboost algorithm is the selection of a weighted set of several classifiers as the final classification. The objective function of the Adaboost-DLDA (ABDLDA) method is defined as:

$$J = \sum_{i=1}^{n} e^{-y_i F(x_i)} = \sum_{i=1}^{n} \exp\left(-y_i \sum_{t=1}^{T} \omega_t f_t(x_i) \right). \tag{38}$$

where x_i and y_i represent the training sample and its class label, respectively. Proposed ABDLDA method classifies a nonlinear data set by combining several linear classifiers. Also, the sample density is calculated locally; since in each step, only the density of the samples that selected by roulette wheel (Back, 1996) is calculated. Thus, the training time is reduced, and it is suitable for big data classification.

First, in the training phase, all training samples' weight is initialized with a uniform distribution to provide an equal chance to choose each training sample. Then, some training samples are selected based on weighted distribution using the roulette wheel algorithm. The basic DLDA classifier is applied to this subset, and then the classifier error is calculated. If the classifier $f_t(x_i)$ has a 50% or higher error, this classifier will be ignored; otherwise, the value of ω_t is obtained, which determines the accuracy of this classifier. The weight of samples that have not correctly classified is increased, and other samples' weight is decreased. These steps are repeated until the final classifier's error (derived from a weighted combination of several classifiers) reaches an acceptable level.

The proposed Adaboost-Weighted-DLDA (ABWDLDA) method is appropriate for classifying the nonlinear data set. In the training phase of ABWDLDA method, first, the Gaussian kernel is inserted on each piece of data. Then, the value of Gaussian kernel allocated to each test sample is calculated, and the result as weight is multiplied by the given data. In this way, the nonlinear data is classified faster. Like the proposed ABDLDA method, the basis classifiers are built, where the total error is smaller than the small positive integer value (ε). ABDLDA and ABWDLDA methods do not require a basic assumption of the data distribution (normal distribution assumptions/single-mode-distributions), and they can classify each data set with any distribution. The proposed ABWDLDA method has more run time for a linear data set than the proposed ABDLDA method and converges late due to calculating the Gaussian kernel for each sample.

4. Experimental results and discussion

The evaluation criteria are introduced in this section, and then the data sets – collected from the UCI Machine Learning Repository[1] – are presented. In our simulation, data fragmentation to a pair of learning/evaluation sets is performed using a 10-fold-cross validated paired t-Test. The proposed methods (DLDA, MDLDA, ABDLDA, ABWDLDA) are compared with several methods, i.e. Orthogonal LDA (OLDA) (Ye & Yu, 2005), Uncorrelated LDA (ULDA) (Ye & Yu, 2005), LODA (Zhang & Chow, 2012), Null Space LDA (NLDA) (Chen et al., 2000), Norm1 LDA (LDA-L1) (Wang et al., 2013), Regularized LDA (RLDA) (Guo et al., 2007), LDA, L1BLDA (Li et al., 2019), L2BLDA (Li et al., 2019), and RNNL2BLDA (Guo et al., 2021). Also, the proposed methods were compared with LDA and multi-class LODA to demonstrate the multi-class capability.

4.1. Evaluation criteria

Accuracy (ACC): It is the rate of samples that are accurately predicted:

$$ACC = \frac{TN + TP}{FP + FN + TP + TN}. \tag{39}$$

Error: The rate of samples that are falsely predicted:

$$Error = \frac{FN + FP}{FP + FN + TP + TN} = 1 - ACC. \tag{40}$$

Detection Rate (DR): This is the ratio between the number of correctly detected attacks and the total number of attacks and is defined as:

$$DR = \frac{TP}{\text{Number of targets}}. \tag{41}$$

False Alarm Rate (FAR): This criterion is also defined to determine the final performance of the intrusion detection algorithms and obtained as:

$$FAR = \frac{FP}{\text{Number of outliers}}. \tag{42}$$

Friedman test: It is used to compare and identify the difference between several classifiers' performance simultaneously (Bahraini, Ghazi,

[1] http://archive.ics.uci.edu/ml/datasets.html

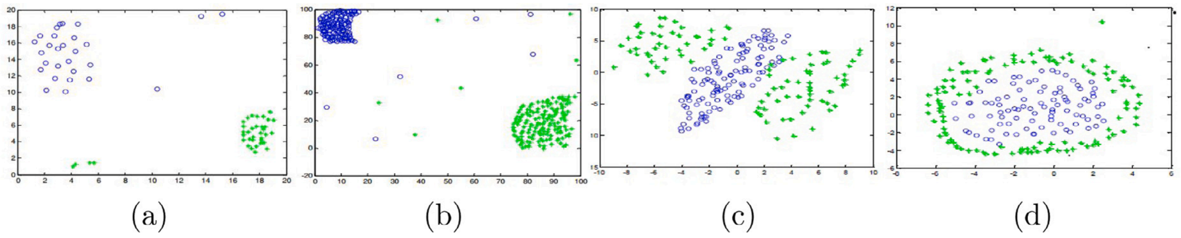


Fig. 1. Synthetic data sets, (a) Out1, (b) Out2, (c) Multi1, (d) Multi2.

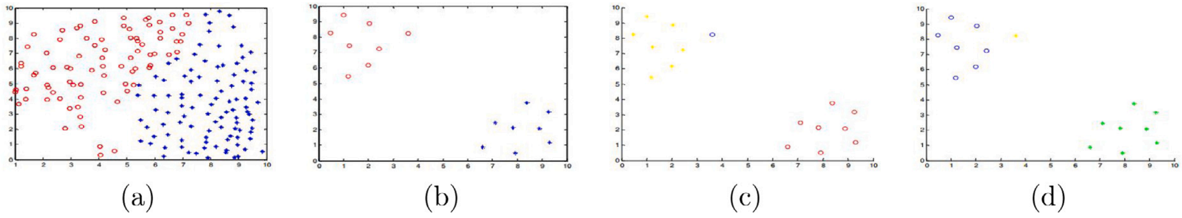


Fig. 2. Output of proposed DLDA method on synthetic Singular data set, indicated solving the singularity problem, (a) Training set, (b) Test set, (c) DLDA without regularized term ($\alpha = 0$), error rate = 0.93%, (d) DLDA with regularized term ($\alpha \neq 0$), error rate = 0.06%.

Table 1
Real data sets.

Data set	Feature Num.	Sample Num.	Classes Num.
Aggregation	2	788	2
Banknote	2	1372	2
Bridge	2	232	2
Compound	2	399	2
Flame	2	240	2
Jain	2	373	2
Spiral	2	312	2
TwoDiamonds	2	400	2
Iris	4	150	3
Glass	9	214	6
Sonar	60	208	2
Liver	6	245	2
Ionosphere	34	351	2
Splice	60	1000	2
Cloud	10	2948	2

Table 2
Real NSL-KDD data set.

Classes	Normal	PROBE	R2L	U2R	DOS	Sum of samples
NSL-KDD Train+	67343	11656	995	52	45927	125973
NSL-KDD Test+	9710	2422	2887	67	7458	22544

Table 3
ACC for proposed DLDA method on real two-dimensional UCI Data Sets: Aggregation, Bridge, Compound, Flame, Jain, Spiral, TwoDiamonds, and Banknote.

Data sets	Aggregation	Banknote	Bridge	Compound	Flame	Jain	Spiral	Spiral
ACC(%)	76.79	55.03	100	93.98	85.42	94.64	72.44	100

& Yazdi, 2020; Demšar, 2006). First, the performance of all algorithms is calculated and then arranged in descending order. After this, Friedman statistic parameter $\mathcal{X}_F^2$ is calculated as follows:

$$\mathcal{X}_F^2 = \frac{12N}{k(k+1)} \left(\sum_{\langle j \rangle} R_j^2 - \frac{k(k+1)^2}{4} \right). \tag{43}$$

where, $R_j = \frac{1}{N} \sum_{\langle i \rangle} r_i^j$ is the average rank of each algorithm, and r_i^j represents the rank of the jth algorithm on the ith data set. k and N represent the number of compared algorithms and data sets, respectively. If this statistic is greater than the corresponding number in the chi-square distribution table with the degree of freedom $k - 1$, the overall significance of the difference between the algorithms will be proved (Lehmann & Romano, 2006).

4.2. Data sets

The experiments have been carried out on a collection of synthetic and real data sets to demonstrate proposed algorithms' performance, i.e., DLDA, MDLDA, ABWDLDA, and ABDLDA.

The synthetic data sets: The synthetic data sets with different distributions and complexities are presented for evaluating various parts of simulations. For example, we generated and used a synthetic data set called singular for singularity problem experiments. In this data set, the number of samples is 50 with dimensions 100, so that the singularity problem can occur. As you can see in Fig. 1, out1 and out2 data are used for the experiments of outliers problem. Multi1 and Multi2 data sets have been used in our simulations to test the ability of classifying the nonlinear and multi-mode data set. Real data sets: Real UCI data sets used in our simulations are presented in Table 1 with the numbers of samples, features and classes. For another example of real data set, the NSL-KDD data set is used to evaluate the proposed method for intrusion detection in computer networks (Tavallaee, Bagheri, Lu, & Ghorbani, 2009). The characteristics of this data are summarized in Table 2. There are 41 features for each sample of the NSL-KDD. There are four different attack groups in this data set, including PROBE,[2] R2L,[3] U2R,[4] and DOS[5] and should be distinguished from the normal users.

4.3. Results and discussion

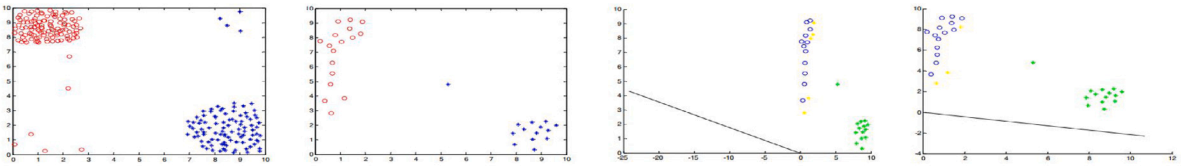
The singularity problem in LDA method is solved using the proposed DLDA method, as shown in Fig. 2. According to this figure, the regularization parameter α in Eq. (9) adjusts DLDA to be generalizable. The synthetic data in Fig. 2 is investigated in the two cases; without regularized term ($\alpha = 0$) and with regularized term ($\alpha = 0.05$). In these

[2] Probing: The attacker explores the network to collect information and identify weak points.

[3] Remote-to-Local:The attacker logged in as a user by sending data packets through the web and accesses it locally.

[4] User-to-Root: The attacker first accesses the system user ID, then exploits the vulnerable points to reach the system root.

[5] Denial-Of-Service: attacker completely interrupts system performance by keeping busy various system resources.



(a) (b) (c) (d)

Fig. 3. Effect of the outliers in performance of LDA and proposed DLDA method, (a) Training set, (b) Test set, (c) The output of LDA method, error rate = 0.023%, (d) Output of proposed DLDA method, error rate = 0.014%.

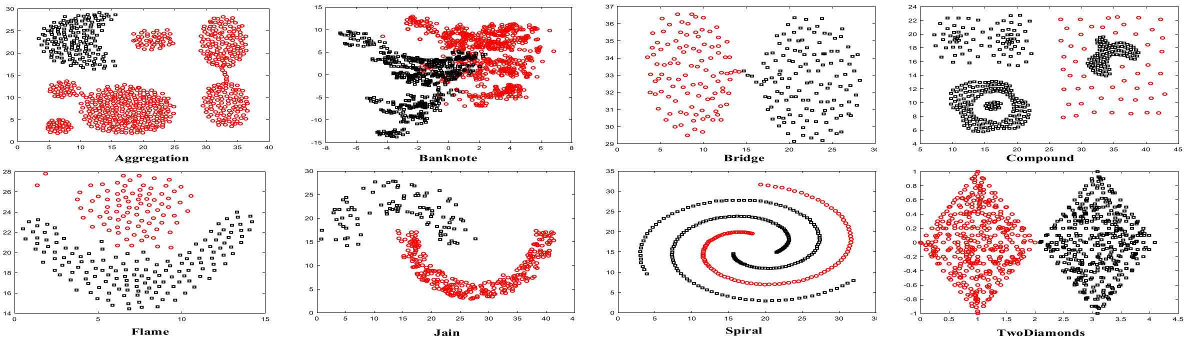


Fig. 4. Real two-dimensional UCI data sets: Aggregation, Bridge, Compound, Flame, Jain, Spiral, TwoDiamonds, and Banknote.

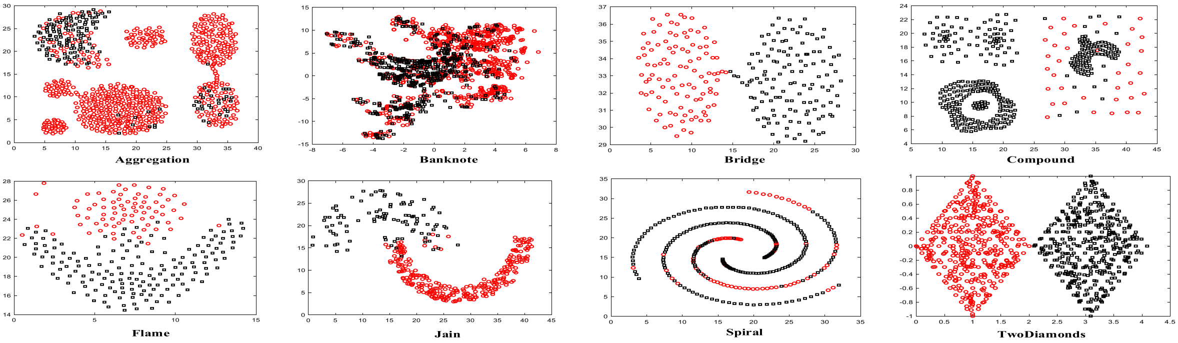


Fig. 5. Results of proposed DLDA method to classify real two-dimensional UCI data sets: Aggregation, Bridge, Compound, Flame, Jain, Spiral, TwoDiamonds, and Banknote.

cases, the error rates are 0.93% and 0.06%, respectively. If value α is set to 0.5, the classification error is 0%. With the correct setting of this parameter, we can achieve the acceptable performance of the proposed method.

In Fig. 3, to study the effect of the outliers in the proposed DLDA method's performance, we used a synthetic data set with two classes and 210 samples (both classes have outliers). Sub-figures (a) and (b) show the training and the test sets, where data includes 31 samples in the test step. As you can see in (c) and (d), the separator line varies in two methods, and due to the effect of outliers on performance, the DLDA classifier has performed better than the LDA, and the errors are 0.014% and 0.023%, respectively.

We also used 8 real two-dimensional UCI data sets, called Aggregation, Bridge, Compound, Flame, Jain, Spiral, TwoDiamonds, and Banknote (Bahraini et al., 2020), which are shown in Fig. 4 and described in Table 1. These data sets are two-dimensional in two classes. According to the results reported in Fig. 5, the proposed DLDA method classifies these data sets into two classes. The ACC values for each data set also are summarized in Table 3. As you can see, the proposed DLDA method works completely error-free for both Bridge and Spiral data. It also has acceptable performance for both data Compound and Jain with ACC values of 93.98% and 94.64%, respectively. The proposed DLDA method's performance for the three data sets, Aggregation, Banknote, and Spiral, has reached 76.79, 55.03, and 72.44, respectively (due to the complex structure of these three data sets). We take these eight real two-dimensional UCI data sets as examples to study the effect of two parameters α and h in the convergence of cost function J and ACC of the proposed DLDA method reported in Figs. 6 and 7, respectively. In Fig. 6, the values of J are obtained in term of α and h. As shown in these results, the minimum value of J for the proposed DLDA method for each of these 8 data sets occurs at a unique value of α and h. Accordingly, these two values must be optimally set for each data set. Also, in Fig. 7, the values of ACC are plotted in terms of α and h. It can be said that the highest accuracy of classifying each data set occurs on a unique α and h for the proposed DLDA method. For example, as shown in Figs. 6 and 7, for Banknote data set the minimum value of $J = 0.003227$ and the maximum of $ACC = 59.69$ occur at $\alpha = 0.995$, $h = 9.6$ and $\alpha = 0.0175$, $h = 0$, respectively. These results led us to optimally find these unique values for each data by grid search method.

We performed some experiments according to Fig. 8 to show the performance of two proposed methods ABDLDA and ABWDLDA in the face of multi-mode and nonlinear data sets. The synthetic Multi1 data is well classified by these two methods. The classifiers' composition continues until the number of wrong classified samples reaches approximately 0.1% of the total number of samples.

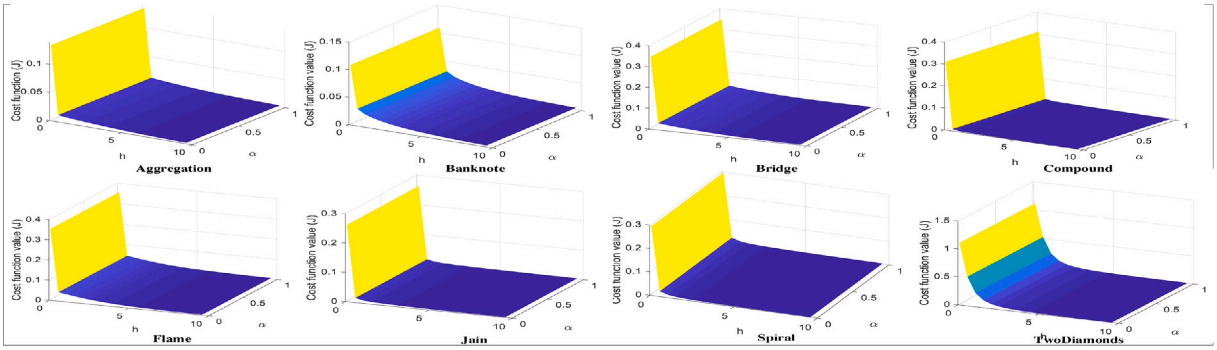


Fig. 6. J values of proposed DLDA method on Real two-dimensional UCI data sets: Aggregation, Bridge, Compound, Flame, Jain, Spiral, TwoDiamonds, and Banknote.

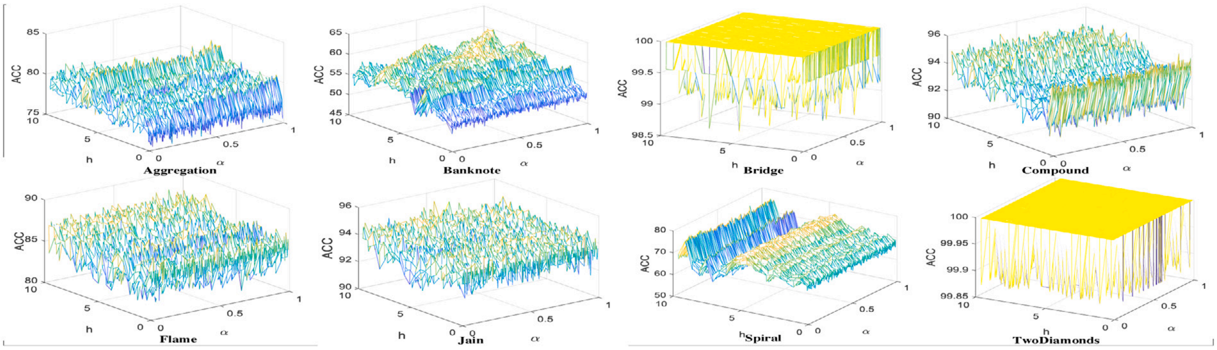


Fig. 7. ACC Results of proposed DLDA method to classify real two-dimensional UCI data sets: Aggregation, Bridge, Compound, Flame, Jain, Spiral, TwoDiamonds, and Banknote.

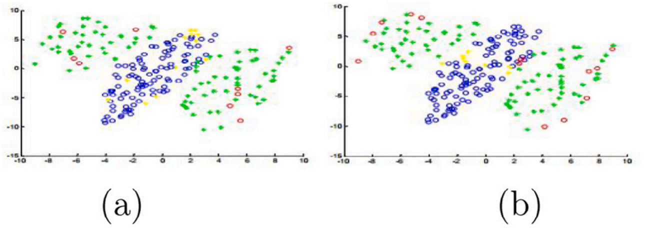


Fig. 8. Classification of synthetic Multi1 data set (as multi-mode and nonlinear data set), (a) Output of proposed ABDLDA method, training error and time 2.94% and 66.86 s, (b) Output of proposed ABWDLDA method, training error and time 3.04% and 28.36 s.

Table 4
ACC for proposed ABDLDA, Inc-SVDD and LIBSVM on NSL-KDD data set.

		Inc-SVDD	LIBSVM	Proposed ABDLDA
Classes	Normal	**93.44**	92.36	87.30
	DOS	82.74	81.19	**91.66**
	PROBE	87.58	80.32	**92.31**
	R2L	76.41	78.20	**88.64**
	U2R	89.00	88.30	**98.73**

Also, the proposed methods ABDLDA and ABWDLDA were compared with the accelerated classifiers LIBSVM (Chang & Lin, 2011), and Incremental Support Vector Data Description (Inc-SVDD) (Hua & Ding, 2011) to compare their performance to classify the big data sets with two classes.

In Table 4, the results of the proposed ABDLDA method are compared with Inc-SVDD and LIBSVM methods on the NSL-KDD data set, which was online as big data sets with two classes (in this test, each NSL-KDD data set class is considered the target class and other classes have regarded as the attack class). As shown in these results, the proposed ABDLDA method has the highest accuracy for all other classes except for the normal class. The number of selected train samples from each class is assumed to be equal to train the basic classifiers in the ABDLDA and ABWDLDA methods. In Table 5, DR and FAR criteria are presented on the NSL-KDD data set for each attack class. As you can see, the proposed ABDLDA method has the lowest FAR rate in all classes, also the highest value of DR for the DOS class.

Fig. 9 shows the ACC criterion in the training sets of the synthetic data sets (a) Multi1 and Multi2, (b) Singular and Out2, and (c) Out1

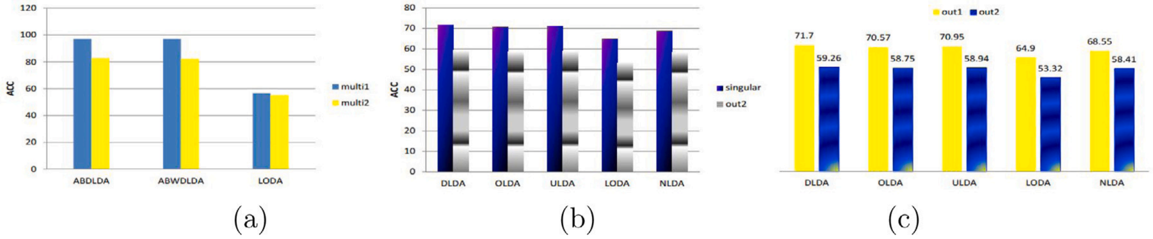


Fig. 9. ACC for synthetic data sets, (a) ACC of the proposed algorithms, ABDLDA, ABWDLDA and LODA, for non-linear and multi-mode synthetic data sets, Multi1, Multi2, (b) ACC on the synthetic data sets, Singular and Out2, (b) ACC on the synthetic data sets, Out1 and Out2 with outlier samples.

Table 5
DR, FAR criteria for the proposed ABDLDA method, Inc-SVDD and LIBSVM on the real NSL-KDD data set.

		DR			FAR		
		Inc-SVDD	LIBSVM	Proposed ABDLDA	Inc-SVDD	LIBSVM	Proposed ABDLDA
Classes	DOS	0.77	0.75	**0.79**	0.23	0.19	**0.02**
	PROBE	**0.89**	0.88	0.77	0.1	0.8	**0.05**
	R2L	0.81	**0.83**	0.80	0.021	0.22	**0.002**
	U2R	0.88	**0.9**	0.50	0.12	0.1	**0.005**

Table 6
Results of LODA, proposed ABDLDA and ABWDLDA methods on multi-mode synthetic data sets: Multi1, Multi2.

Synthetic Data sets	Algorithms	LODA	Proposed ABWDLDA	Proposed ABDLDA
Multi1	Training Error(%)	43.32	3.04	**2.94**
	Training Time(S)	0.207	28.36	66.86
Multi2	Training Error(%)	44.81	17.69	**17.02**
	Training Time(S)	0.49	73.64	234.58

Table 7
Results in dealing with the singularity and outliers problems in two synthetic data sets: singular and Out2.

Synthetic Data sets	Algorithms	OLDA	ULDA	NLDA	RLDA	Proposed DLDA
Singular	Training Error(%)	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	Training Time(S)	0.003	0.002	0.16	0.006	0.02
Out2	Training Error(%)	2.96	3.17	2.56	2.56	**1.88**
	Testing Error(%)	7.93	12.33	**6.80**	**6.80**	**6.80**
	Training Time(S)	0.001	0.001	0.001	0.001	0.07

and Out2, for the proposed and other classifiers. As shown in Fig. 9-(a), two proposed methods ABDLDA and ABWDLDA have higher ACC values than the proposed LODA method to classify the Multi1 and Multi2 Data sets. In Fig. 9-(b), the proposed DLDA method is compared to four methods, OLDA, ULDA, LODA, and NLDA. You can see the proposed DLDA has the highest ACC than others with a value equal to 72.45%. Also, in Fig. 9-(c) ACC is calculated for these methods on the synthetic Out1 and Out2 data sets. For these data sets, the proposed DLDA method has a better performance and has reached the value of ACC = 71.7%.

The ABDLDA and ABWDLDA are compared with the multi-mode LODA classifier on two synthetic Multi1 and Multi2 data sets. According to the summarized results in Table 6, the training and testing errors of ABDLDA and ABWDLDA methods are much less than the LODA, but their training time is longer.

In this experiment, the basis classifiers are created as long as the number of incorrect classified samples exceeds 0.1% of the total number of samples.

Table 7 shows the training error, training time, and testing error of the proposed DLDA method and the comparison methods on the Singular and Out2 synthetic data sets. The test data are also synthetic for these two data sets, and they have only an experimental aspect. The training error on the Singular data set is zero for all classifiers. The proposed DLDA method's training error on the Out2 data set is the lowest value, which is also due to the low impact of the outliers in its performance. The testing error for the proposed DLDA, RLDA, and NLDA methods is at the lowest level and is at the highest level for the ULDA classifier. The proposed DLDA method's training time is more than the other classifiers due to calculating the density of samples $p(x_i)$.

The classification results for two-class data sets are shown in Table 8. As you can see in this table, in most data sets, the proposed DLDA method has fewer errors and higher ACC values than others. And the Average Rank of this method is better than other comparison methods, which is equal to 2.5. From the results of Friedman's test, there is a significant difference between the classifiers with a 99.5% level of confidence. The next position is related to RNNL2BLDA method, which

Table 8
Results of real two-class data sets i.e. Sonar, Liver, Ionosphere, Splice, and Cloud for NLDA, OLDA, ULDA, LDA_L1, LODA, LDA, L1BLDA, L2BLDA, RNNL2BLDA, and proposed DLDA methods.

Real Data sets	Algorithms	NLDA	OLDA	ULDA	LDA_L1	LODA	LDA	L1BLDA	L2BLDA	RNNL2BLDA	Proposed DLDA
Sonar	Training Time(S)	0.0016	0.0014	0.0013	0.9078	0.1684	0.0013	**0.0011**	**0.0011**	0.0016	0.045
	Training Error(%)	26.20	**23.78**	25.89	26.17	35.10	26.20	27.32	26.20	24.56	24.18
	Testing Error(%)	30.00	30.76	31.43	33.02	33.98	30.00	33.45	30.00	30.13	**28.52**
	ACC	68.55(6)	70.57(3)	70.95(2)	67.50(8)	64.90(10)	68.55(6)	67.02(9)	68.55(6)	70.43(4)	**71.70(1)**
Liver	Training Time(S)	0.0019	**0.0018**	0.0019	0.208	0.417	0.0019	0.0019	0.0019	0.0023	0.106
	Training Error(%)	45.37	37.37	38.08	**37.28**	42.13	45.37	38.39	37.70	38.17	38.24
	Testing Error(%)	40.63	40.81	40.82	40.35	44.54	40.63	41.75	40.53	39.62	**38.82**
	ACC	58.41(6.5)	58.75(4)	58.94(2)	57.91(8)	53.32(10)	58.41(6.5)	57.81(9)	58.73(5)	58.89(3)	**59.26(1)**
Ionosphere	Training Time(S)	0.0059	0.0072	0.0074	0.193	0.468	0.0044	**0.0040**	0.0043	0.0044	0.27
	Training Error(%)	35.16	12.20	12.20	11.97	34.35	12.20	12.00	11.32	11.20	**11.18**
	Testing Error(%)	31.47	15.85	15.85	17.62	32.37	16.36	17.73	17.60	**15.80**	17.11
	ACC	69.18(9)	**85.11(1)**	84.77(2)	84.44(4)	67.40(10)	83.92(8)	84.18(7)	84.23(6)	84.37(5)	84.56(3)
Splice	Training Time(S)	0.008	0.042	0.039	3.97	3.39	0.005	**0.001**	**0.001**	0.003	0.82
	Training Error(%)	46.72	46.20	47.62	47.68	48.70	46.72	46.39	46.30	46.01	**45.94**
	Testing Error(%)	**44.50**	48.38	48.25	46.63	48.63	**44.50**	48.19	44.92	**44.50**	47.13
	ACC	**55.50(2)**	51.63(9)	51.75(8)	53.38(5)	51.38(10)	**55.50(2)**	52.50(7)	55.07(4)	**55.50(2)**	52.88(6)
Cloud	Training Time(S)	0.014	0.159	0.158	0.568	14.96	0.0107	**0.0070**	**0.0070**	**0.0070**	3.22
	Training Error(%)	**0.05**	**0.00**	**0.00**	0.16	0.70	0.05	0.08	**0.00**	**0.00**	**0.00**
	Testing Error(%)	0.05	0.05	0.05	0.39	0.83	0.05	0.20	0.05	**0.00**	**0.00**
	ACC	99.95(5)	99.95(5)	99.95(5)	99.61(9)	99.17(10)	99.95(5)	99.76(8)	99.95(5)	**100(1.5)**	**100(1.5)**
Average Rank		5.7	4.4	3.8	6.8	10	5.5	8	5.2	3.1	**2.5**
$\mathcal{X}_F^2$		25.7345									
Chi-Sq (from table dis.)		23.5890									
P-Value $Prob > Chi - Sq$		0.005									

Table 9
Results of Real Multi-class Iris and Glass Data Sets for NLDA, OLDA, ULDA, LDA_L1, LODA, LDA, L1BLDA, L2BLDA, RNNL2BLDA, and Proposed MDLDA.

Synthetic Data sets	Algorithms	NLDA	OLDA	ULDA	LDA_L1	LODA	LDA	L1BLDA	L2BLDA	RNNL2BLDA	Proposed MDLDA
Iris	Training Time(S)	0.0009	0.0005	0.0005	1.62	0.054	0.0009	**0.0003**	0.0004	0.0005	0.037
	Training Error(%)	10.00	**4.13**	25.20	16.13	8.27	26.53	10.00	7.82	5.60	**4.13**
	Testing Error(%)	8.27	5.07	30.27	23.07	7.33	25.07	8.69	8.10	6.38	**4.00**
	ACC	68.55(7)	70.57(4)	70.95(2)	67.5(9)	64.9(10)	68.55(7)	68.55(7)	68.79(5)	70.89(3)	**71.70(1)**
Glass	Training Time(S)	**0.001**	0.002	**0.001**	0.008	0.08	0.0014	**0.001**	0.0014	0.0014	0.131
	Training Error(%)	45.09	**32.56**	40.13	52.89	45.44	44.02	49.09	40.15	40.10	33.13
	Testing Error(%)	44.55	32.75	32.27	51.84	53.31	39.76	44.55	32.30	32.18	**31.84**
	ACC	58.41(7.5)	58.75(5)	58.94(3)	57.91(9)	53.32(10)	58.41(7.5)	58.41(6)	58.93(4)	59.00(2)	**59.26(1)**
Average Rank		7.25	4.5	2.5	9	10	7.25	6.5	4.5	2.5	**1**
$\mathcal{X}_F^2$		17.4272									
Chi-Sq (from table dis.)		16.9190									
P-Value $Prob > Chi - Sq$		0.050									

Table 10
Computational complexity for NLDA, OLDA, ULDA, LDA_L1, LODA, LDA, L1BLDA, L2BLDA, RNNL2BLDA, Proposed DLDA, and Proposed MDLDA.

Method	Computing complexity
NLDA	$O(4d^2n)$
OLDA	$O(14dn^2 + 4dnc + 2dc^2)$
ULDA	$O(14dn^2 + 4dnc)$
LDA_L1	$O(T(n + c)d)$
LODA	$O((n + c + 1)d + c^2)$
LDA	$O((n + c + 1)d)$
L1BLDA	$O(d^3)$ or $O(d^3 + Td^2l)$
L2BLDA	$O(d^3)$
RNNL2BLDA	$O(d^3 + nlogn)$
Proposed DLDA	$O(dn^2 + d^2n + n^2)$
Proposed MDLDA	$O(c^2 \log c(dn^2 + d^2n + n^2))$

is ranked 3.1. According to the results reported in this table, methods L1BLDA and L2BLDA on most of the data sets have the fastest results.

The proposed MDLDA method is compared with the multi-class classifiers, such as the standard multi-class LDA, NLDA, OLDA, ULDA, LDA_L1, and LODA. To calculate the test error using the K-fold method, a part of the data is given to the classifiers as training, and the other part is considered the test set. According to the results of Table 9, the proposed MDLDA classifier on the Iris data set has the lowest training and test errors. Furthermore, the training error of OLDA in the Glass data set with six classes is less than the others, and the proposed MDLDA also has the lowest test error in this data set. Also, the Friedman test was used to investigate the significant difference in the algorithms' performance. In these results, the parameter $\mathcal{X}_F^2$ is obtained with a value of 17.4272. The chi-square value in its table with a degree of freedom 9 and a significant level of 0.050 is equivalent to 16.9190. So, with 95% confidence, we can say that there is a significant difference between the classifiers' performances. After that, ULDA and RNNL2BLDA have acceptable performance and got the rank of 2.5. Also, L1BLDA with a rank of 6.5 is the fastest method.

To better evaluate the speed of the compared methods and the proposed methods, we extracted the computational complexity of all of them. These results were summarized in Table 10. According to this, L1BLDA and L2BLDA seem to be the fastest methods. Also, the methods in which n^2 has appeared in computational complexity order are among the slowest.

5. Conclusion

The proposed DLDA classifier was presented to solve some of the LDA problems, like the reduced impact of the outlier samples on the classification performance. For this purpose, the density of samples interfered in classification, and Parzen window was used to estimate this density. Since the outliers have a low density, they do not affect much on the classification performance. One of the notable limitations of the proposed DLDA method is determining the precise value of the Parzen window width parameter (h), which directly impacts the performance of the proposed DLDA and is estimated by using the Grid search method for each data set. One of the other proposed methods' challenges is that these take a long time to calculate the density of samples in the big data sets due to the computational time with order $O(dn^2 + d^2n + n^2)$, which is dependent on data size n. The classifiers combination technique such as Adaboost was used in this paper to classify the big data set, so two proposed ABDLDA and ABWDLDA methods were introduced. The samples' density $p(x_i)$ is calculated locally in the proposed ABDLDA and ABWDLDA methods to solve this. One can adopt the new strategies to solve these two limitations, reduce the run time, and improve performance in future work.

CRediT authorship contribution statement

Tahereh Bahraini: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft. **Seyed Mohammad Hosseini:** Formal analysis, Writing – original draft, Investigation. **Mahbubeh Ghasempour:** Methodology, Software. **Hadi Sadoghi Yazdi:** Conceptualization, Investigation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alimoglu, F., & Alpaydin, E. (1997). Combining multiple representations and classifiers for pen-based handwritten digit recognition. *Vol. 2*, In *Proceedings of the fourth international conference on document analysis and recognition* (pp. 637–640). IEEE.

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press.

Bahraini, T., Ghazi, S., & Yazdi, H. S. (2020). Toward optimum fuzzy support vector machines using error distribution. *Engineering Applications of Artificial Intelligence*, *90*, 103–545.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720.

Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Cai, J., & Huang, X. (2018). Modified sparse linear-discriminant analysis via nonconvex penalties. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(10), 4957–4966.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 1–27.

Chen, Y., & Jin, Z. (2012). Reconstructive discriminant analysis: A feature extraction method induced from linear regression classification. *Neurocomputing*, *87*, 41–50.

Chen, L.-F., Liao, H.-Y. M., Ko, M.-T., Lin, J.-C., & Yu, G.-J. (2000). A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, *33*(10), 1713–1726.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (second ed.). (p. 35). New York, USA: John Wiley&Sons.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society for Artificial Intelligence*, *14*(771–780), 1612.

Gao, J., Wang, Q., & Yuan, Y. (2019). SCAR: Spatial-/channel-wise attention regression networks for crowd counting. *Neurocomputing*, *363*, 1–8.

Guo, Y.-R., Bai, Y.-Q., Li, C.-N., Shao, Y.-H., Ye, Y.-F., & Jiang, C.-z. (2021). Reverse nearest neighbors Bhattacharyya bound linear discriminant analysis for multimodal classification. *Engineering Applications of Artificial Intelligence*, *97*, Article 104033.

Guo, Y., Hastie, T., & Tibshirani, R. (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, *8*(1), 86–100.

Hua, X., & Ding, S. (2011). Incremental learning algorithm for support vector data description. *JSW*, *6*(7), 1166–1173.

Jia, Y., Ma, J., & Gan, L. (2017). Combined optimization of feature reduction and classification for radiometric identification. *IEEE Signal Processing Letters*, *24*(5), 584–588.

Jia, J., Ruan, Q., & Jin, Y. (2016). Geometric preserving local fisher discriminant analysis for person re-identification. *Neurocomputing*, *205*, 92–105.

Lehmann, E. L., & Romano, J. P. (2006). *Testing statistical hypotheses*. Springer Science & Business Media.

Li, C.-N., Shao, Y.-H., Wang, Z., Deng, N.-Y., & Yang, Z.-M. (2019). Robust bhattacharyya bound linear discriminant analysis through an adaptive algorithm. *Knowledge-Based Systems*, *183*, Article 104858.

Li, D., Wang, L., Wang, J., Xue, Z., & Wong, S. T. (2017). Transductive local fisher discriminant analysis for gene expression profile-based cancer classification. In *2017 IEEE EMBS international conference on biomedical & health informatics (BHI)* (pp. 49–52). IEEE.

Li, P., Zhou, W., Huang, X., Zhu, X., Liu, H., Ma, T., et al. (2018). Improved graph embedding for robust recognition with outliers. *Scientific Reports*, *8*(1), 1–11.

Liu, W., Pokharel, P. P., & Principe, J. C. (2007). Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Transactions on Signal Processing*, *55*(11), 5286–5298.

Nie, F., Wang, Z., Wang, R., & Li, X. (2019). Submanifold-preserving discriminant analysis with an auto-optimized graph. *IEEE Transactions on Cybernetics*, *50*(8), 3682–3695.

Nie, F., Wang, Z., Wang, R., Wang, Z., & Li, X. (2020). Adaptive local linear discriminant analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *14*(1), 1–19.

Peng, X., Qiao, Y., Peng, Q., & Wang, Q. (2014). Large margin dimensionality reduction for action similarity labeling. *IEEE Signal Processing Letters*, *21*(8), 1022–1025.

Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *10*(2), 159–193.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, *37*(3), 297–336.

Tang, E. K., Suganthan, P. N., Yao, X., & Qin, A. K. (2005). Linear dimensionality reduction using relevance weighted LDA. *Pattern Recognition*, *38*(4), 485–493.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1–6). IEEE.

Wan, H., Wang, H., Guo, G., & Wei, X. (2017). Separability-oriented subclass discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(2), 409–422.

Wang, Q., Gao, J., & Yuan, Y. (2017). Embedding structured contour and location prior in siamesed fully convolutional networks for road detection. *IEEE Transactions on Intelligent Transportation Systems*, *19*(1), 230–241.

Wang, H., Lu, X., Hu, Z., & Zheng, W. (2013). Fisher discriminant analysis with L1-norm. *IEEE Transactions on Cybernetics*, *44*(6), 828–842.

Wang, Z., Ruan, Q., & An, G. (2016). Facial expression recognition using sparse local Fisher discriminant analysis. *Neurocomputing*, *174*, 756–766.

Wen, J., Fang, X., Cui, J., Fei, L., Yan, K., Chen, Y., et al. (2018). Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(2), 390–403.

Yang, J., Yang, J., & Zhang, D. (2008). Median Fisher discriminator: a robust feature extraction method with applications to biometrics. *Frontiers of Computer Science in China*, *2*(3), 295–305.

Ye, J. (2007). Least squares linear discriminant analysis. In *Proceedings of the 24th international conference on machine learning* (pp. 1087–1093).

Ye, J., & Ji, S. (2010). Discriminant analysis for dimensionality reduction: An overview of recent developments. In *Biometrics: theory, methods, and applications*. New York: Wiley-IEEE Press, Citeseer.

Ye, Q., Yang, J., Liu, F., Zhao, C., Ye, N., & Yin, T. (2016). L1-norm distance linear discriminant analysis based on an effective iterative algorithm. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(1), 114–129.

Ye, J., & Yu, B. (2005). Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, *6*(4).

Ye, Q., Zhao, H., Fu, L., & Gao, S. (2018). Underlying connections between algorithms for nongreedy LDA-L1. *IEEE Transactions on Image Processing*, *27*(5), 2557–2559.

Yu, H., Gao, L., Li, W., Du, Q., & Zhang, B. (2017). Locality sensitive discriminant analysis for group sparse representation-based hyperspectral imagery classification. *IEEE Geoscience and Remote Sensing Letters*, *14*(8), 1358–1362.

Zhang, Z., & Chow, T. W. (2012). Robust linearly optimized discriminant analysis. *Neurocomputing*, *79*, 140–157.

Zhao, B., Li, X., & Lu, X. (2019). CAM-RNN: Co-attention model based RNN for video captioning. *IEEE Transactions on Image Processing*, *28*(11), 5552–5565.

Zhao, B., Li, X., & Lu, X. (2020). TTH-RNN: Tensor-train hierarchical recurrent neural network for video summarization. *IEEE Transactions on Industrial Electronics*, *68*(4), 3629–3637.

Zhong, F., & Zhang, J. (2013). Linear discriminant analysis based on L1-norm maximization. *IEEE Transactions on Image Processing*, *22*(8), 3018–3027.

Zhou, W., Li, P., Wang, X., Li, F., Liu, H., Zhang, R., et al. (2015). Lp norm spectral regression for feature extraction in outlier conditions. In *2015 IEEE international conference on digital signal processing (DSP)*. IEEE.