Expert Systems With Applications 79 (2017) 44–52

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Linear classifier design under heteroscedasticity in Linear Discriminant Analysis

Kojo Sarfo Gyamfi*, James Brusey, Andrew Hunt, Elena Gaura

Faculty of Engineering and Computing, Coventry University, Coventry, CV1 5FB, United Kingdom

ARTICLE INFO

Article history:
Received 12 November 2016
Revised 23 February 2017
Accepted 24 February 2017
Available online 24 February 2017

Keywords:
LDA
Heteroscedasticity
Bayes error
Linear classifier

ABSTRACT

Under normality and homoscedasticity assumptions, Linear Discriminant Analysis (LDA) is known to be optimal in terms of minimising the Bayes error for binary classification. In the heteroscedastic case, LDA is not guaranteed to minimise this error. Assuming heteroscedasticity, we derive a linear classifier, the Gaussian Linear Discriminant (GLD), that directly minimises the Bayes error for binary classification. In addition, we also propose a local neighbourhood search (LNS) algorithm to obtain a more robust classifier if the data is known to have a non-normal distribution. We evaluate the proposed classifiers on two artificial and ten real-world datasets that cut across a wide range of application areas including handwriting recognition, medical diagnosis and remote sensing, and then compare our algorithm against existing LDA approaches and other linear classifiers. The GLD is shown to outperform the original LDA procedure in terms of the classification accuracy under heteroscedasticity. While it compares favourably with other existing heteroscedastic LDA approaches, the GLD requires as much as 60 times lower training time on some datasets. Our comparison with the support vector machine (SVM) also shows that, the GLD, together with the LNS, requires as much as 150 times lower training time to achieve an equivalent classification accuracy on some of the datasets. Thus, our algorithms can provide a cheap and reliable option for classification in a lot of expert systems.

1. Introduction

In many applications one encounters the need to classify a given object under one of a number of distinct groups or classes based on a set of features known as the feature vector. A typical example is the task of classifying a machine part under one of a number of health states. Other applications that involve classification include face detection, object recognition, medical diagnosis, credit card fraud prediction and machine fault diagnosis.

A common treatment of such classification problems is to model the conditional density functions of the feature vector (Ng & Jordan, 2002). Then, the most likely class to which a feature vector belongs can be chosen as the class that maximises the a posteriori probability of the feature vector. This is known as the maximum a posteriori (MAP) decision rule.

Let K be the number of classes, $\mathcal{C}_k$ be the kth class, $\mathbf{x}$ be a feature vector and $\mathcal{D}_k$ be training samples belonging to the kth class ($k \in \{1, 2, \ldots, K\}$). The MAP decision rule for the classification task is then to choose the most likely class of $\mathbf{x}$, $\mathcal{C}^*(\mathbf{x})$ given as:

$$\mathcal{C}^*(\mathbf{x}) = \arg\max_{\mathcal{C}_k} p(\mathcal{C}_k|\mathbf{x}), \quad k \in \{1, 2, \ldots, K\} \tag{1}$$

We assume for the moment that there are only $K = 2$ classes, i.e. binary classification (we consider multi-class classification in a later section). Then, using Bayes' rule, the two posterior probabilities can be expressed as:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k) \times p(\mathcal{C}_k)}{p(\mathbf{x})}, \quad k \in \{1, 2\} \tag{2}$$

It is often the case that the prior probabilities $p(\mathcal{C}_1)$ and $p(\mathcal{C}_2)$ are known, or else they may be estimable from the relative frequencies of $\mathcal{D}_1$ and $\mathcal{D}_2$ in $\mathcal{D}$ where $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$. Let these priors be given by π_1 and π_2 respectively for class $\mathcal{C}_1$ and $\mathcal{C}_2$. Then, the likelihood ratio defined as:

$$\lambda(\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)} \tag{3}$$

* Corresponding author.
E-mail addresses: gyamfik@uni.coventry.ac.uk (K.S. Gyamfi), j.brusey@coventry.ac.uk (J. Brusey), ab8187@coventry.ac.uk (A. Hunt), csx216@coventry.ac.uk (E. Gaura).

http://dx.doi.org/10.1016/j.eswa.2017.02.039

is compared against a threshold defined as $\tau = \pi_2/\pi_1$ so that one decides on class $\mathcal{C}_1$ if $\lambda(\mathbf{x}) \geq \tau$ and class $\mathcal{C}_2$ otherwise.

Linear Discriminant Analysis (LDA) proceeds from here with two basic assumptions (Izenman, 2009, Chapter 8):

1. The conditional probabilities $p(\mathbf{x}|\mathcal{C}_1)$ and $p(\mathbf{x}|\mathcal{C}_2)$ have multivariate normal distributions.
2. The two classes have equal covariance matrices, an assumption known as homoscedasticity.

Let $\bar{\mathbf{x}}_1$, $\mathbf{\Sigma}_1$ be the mean and covariance matrix of $\mathcal{D}_1$ and $\bar{\mathbf{x}}_2$, $\mathbf{\Sigma}_2$ be the mean and covariance of $\mathcal{D}_2$ respectively. Then, for $k \in \{1, 2\}$,

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{\Sigma}_k)}} \exp\left[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{x} - \bar{\mathbf{x}}_k) \right] \tag{4}$$

where d is the dimensionality of $\mathcal{X}$, which is the feature space of $\mathbf{x}$. Given the above definitions of the conditional probabilities, one may obtain a log-likelihood ratio given as:

$$\ln \lambda(\mathbf{x}) = \frac{1}{2} \ln \frac{\det \mathbf{\Sigma}_2}{\det \mathbf{\Sigma}_1} + \frac{1}{2}\left[(\mathbf{x}-\bar{\mathbf{x}}_2)^T \mathbf{\Sigma}_2^{-1} (\mathbf{x}-\bar{\mathbf{x}}_2) - (\mathbf{x}-\bar{\mathbf{x}}_1)^T \mathbf{\Sigma}_1^{-1} (\mathbf{x} - \bar{\mathbf{x}}_1) \right] \tag{5}$$

which is then compared against $\ln \tau$ so that $\mathcal{C}_1$ is chosen if $\ln \lambda(\mathbf{x}) \geq \ln \tau$, and $\mathcal{C}_2$ otherwise. Thus, the decision rule for classifying a vector $\mathbf{x}$ under class $\mathcal{C}_1$ can be rewritten as:

$$(\mathbf{x} - \bar{\mathbf{x}}_2)^T \mathbf{\Sigma}_2^{-1} (\mathbf{x} - \bar{\mathbf{x}}_2) - (\mathbf{x} - \bar{\mathbf{x}}_1)^T \mathbf{\Sigma}_1^{-1} (\mathbf{x} - \bar{\mathbf{x}}_1) \geq \ln \frac{\tau^2 \det \mathbf{\Sigma}_1}{\det \mathbf{\Sigma}_2} \tag{6}$$

In general, this result is a quadratic discriminant. However, a linear classifier is often desired for the following reasons:

1. A linear classifier is robust against noise since it tends not to overfit (Mika, Ratsch, Weston, Scholkopf, & Mullers, 1999).
2. A linear classifier has relatively shorter training and testing times (Yuan, Ho, & Lin, 2012).
3. Many linear classifiers allow for a transformation of the original feature space into a higher dimensional feature space using the kernel trick for better classification in the case of a non-linear decision boundary (Bishop, 2006, Chapter 6).

By calling on the assumption of homoscedasticity, i.e. $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}_x$, the original quadratic discriminant given by (6) for classifying a given vector $\mathbf{x}$ decomposes into the following linear decision rule:

$$\mathbf{x}^T \mathbf{\Sigma}_x^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} \ln \tau + \frac{1}{2}(\bar{\mathbf{x}}_1^T \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2^T \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_2) \tag{7}$$

Here, $\mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$ is a vector of weights denoted by $\mathbf{w}$ and $\ln \tau + \frac{1}{2}(\bar{\mathbf{x}}_1^T \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2^T \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_2)$ is a threshold denoted by w_0. This linear classifier is also known as Fishers Linear Discriminant. If only the weight vector $\mathbf{w}$ is required for dimensionality reduction, $\mathbf{w}$ may be obtained by maximising Fishers criterion (Fisher, 1936), given by:

$$S = \frac{\mathbf{w}^T (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T \mathbf{w}}{\mathbf{w}^T \mathbf{\Sigma}_x \mathbf{w}} \tag{8}$$

where $\mathbf{\Sigma}_x = n_1 \mathbf{\Sigma}_1 + n_2 \mathbf{\Sigma}_2$ and n_1, n_2 are the cardinalities of $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively.

LDA is the optimal Bayes' classifier for binary classification if the normality and homoscedasticity assumptions hold (Hamsici & Martinez, 2008) (Izenman, 2009, Chapter 8). It demands only the computation of the dot product between $\mathbf{w}$ and $\mathbf{x}$, which is a relatively computationally inexpensive operation.

As a supervised learning algorithm, LDA is performed either for dimensionality reduction (usually followed by classification) (Barber, 2012, Chapter 16; Buturovic, 1994; Duin & Loog, 2004; Sengur, 2008), or directly for the purpose of statistical classification (Fukunaga, 2013, Chapter 4; Izenman, 2009; Mika et al., 1999). LDA has been applied to several problems such as medical diagnosis e.g. Coomans, Jonckheer, Massart, Broeckaert, and Blockx (1978); Polat, Güneş, and Arslan (2008); Sengur (2008); Sharma and Paliwal (2008), face and object recognition e.g. Chen, Liao, Ko, Lin, and Yu (2000); Liu, Chen, Tan, and Zhang (2007); Song, Zhang, Wang, Liu, and Tao (2007); Yu and Yang (2001) and credit card fraud prediction e.g. Mahmoudi and Duman (2015). The widespread use of LDA in these areas is not because the datasets necessarily satisfy the normality and homoscedasticity assumptions, but mainly due to the robustness of LDA against noise, being a linear model (Mika et al., 1999). Since the linear Support Vector Machine (SVM) can be quite expensive to train, especially for large values of K or n ($n = n_1 + n_2$), LDA is often relied upon (Hariharan, Malik, & Ramanan, 2012).

Yet, practical implementation of LDA is not without problems. Of note is the small sample size (SSS) problem that LDA faces with high-dimensional data and much smaller training data (Lu, Plataniotis, & Venetsanopoulos, 2003; Sharma & Paliwal, 2015). When $d \gg n$, the scatter matrix Σ_x is not invertible, as it is not full-rank. Since the decision rule as given by (7) requires the computation of the inverse of Σ_x, the singularity of Σ_x makes the solution infeasible. In works by, for example, Liu et al. (2007); Paliwal and Sharma (2012), this problem is overcome by taking the Moore–Penrose pseudo-inverse of the scatter matrix, rather than the ordinary matrix inverse. Sharma and Paliwal (2008) use a gradient descent approach where one starts from an initial solution of $\mathbf{w}$ and moves in the negative direction of the gradient of Fisher's criterion (8). This method avoids the computation of an inverse altogether. Another approach to solving the SSS problem involves adding a scalar multiple of the identity matrix to the scatter matrix to make the resulting matrix non-singular, a method known as regularised discriminant analysis (Friedman, 1989; Lu et al., 2003).

However, for a given dataset that does not satisfy the homoscedasticity or normality assumption, one would expect that modifications to the original LDA procedure accounting for these violations would yield an improved performance. One such modification, in the case of a non-normal distribution, is the mixture discriminant analysis (Hastie & Tibshirani, 1996; Ju, Kolaczyk, & Gopal, 2003; McLachlan, 2004) in which a non-normal distribution is modelled as a mixture of Gaussians. However, the parameters of the mixture components or even the number of mixture components, are usually not known a priori. Other non-parametric approaches to LDA that remove the normality assumption involve using local neighbourhood structures (Cai, He, Zhou, Han, & Bao, 2007; Fukunaga & Mantock, 1983; Li, Lin, & Tang, 2009) to construct a similarity matrix instead of the scatter matrix Σ_x used in LDA. However, these approaches aim at linear dimensionality reduction, rather than linear classification. Another modification, in the case of a non-linear decision boundary between $\mathcal{D}_1$ and $\mathcal{D}_2$, is the Kernel Fisher Discriminant (KFD) (Mika et al., 1999; Polat et al., 2008; Zhao, Sun, Yu, Liu, & Ye, 2009). KFD maps the original feature space $\mathcal{X}$ into some other space $\mathcal{Y}$ (usually higher dimensional) via the kernel trick (Mika et al., 1999). While the main utility of the kernel is to guarantee linear separability in the transformed space, the kernel may also be employed to transform non-normal data into one that is near-normal.

Our proposed method differs from the above approaches in that we primarily consider violation of the homoscedasticity assumption, and do not address the SSS problem. We seek to provide a linear approximation to the quadratic boundary given by (6) under heteroscedasticity without any kernel transformation; we note

that several heteroscedastic LDA approaches have been proposed to this effect. Nevertheless, for reasons which we highlight in the next section, our contributions in this paper are stated explicitly as follows:

1. We propose a novel linear classifier, which we term the Gaussian Linear Discriminant (GLD), that directly minimises the Bayes error under heteroscedasticity via an efficient optimisation procedure. This is presented in Section 3.
2. We propose a local neighbourhood search method to provide a more robust classifier if the data has a non-normal distribution (Section 4).

2. Related work

Under the heteroscedasticity assumption, many LDA approaches have been proposed among which we mention (Fukunaga, 2013, Chapter 4; Decell & Mayekar, 1977; Decell Jr & Marani, 1976; Duin & Loog, 2004; Loog & Duin, 2002; Malina, 1981; McLachlan, 2004; Zhang & Liu, 2008). As it is known that Fisher's criterion (whose maximisation is equivalent to the LDA derivation described in the Introduction section) only takes into account the difference in the projected class means, existing heteroscedastic LDA approaches tend to obtain a generalisation on Fisher's criterion. In the work of Loog and Duin (2002), for instance, a directed distance matrix (DDM) known as the Chernoff distance, which takes into account the difference in covariance matrices between the two classes as well as the projected class means, is maximised instead of Fisher's criterion (8). The same idea employing the Chernoff criterion is used by Duin and Loog (2004). A wider class of Bregman divergences including the Bhattacharya distance (Decell Jr & Marani, 1976) and the Kullback-Leibler divergence (Decell & Mayekar, 1977) have also been used for heteroscedastic LDA, as Fisher's criterion can be considered a special case of these measures when the covariance matrices of the classes are equal.

However, most of these approaches aim at linear dimensionality reduction, which involves finding a linear transformation that transforms the original data into one of reduced dimensionality, while at the same time maximising the discriminatory information between the classes. Our focus with this paper, however, is not on dimensionality reduction, but on obtaining a Bayes optimal linear classifier for binary classification assuming that the covariance matrices are not equal. As far as we know, the closest work to ours in this regard are the works by Anderson and Bahadur (1962); Fukunaga (2013); Marks and Dunn (1974); Peterson and Mattson (1966)

Obtaining the Bayes optimal linear classifier involves minimising the probability of misclassification p_e as given by:

$$p_e = \pi_1 p(y < w_0|\mathcal{C}_1) + \pi_2 p(y \geq w_0|\mathcal{C}_2) \tag{9}$$

where $y = \mathbf{w}^T\mathbf{x}$. Unfortunately, there is no closed-form solution to the minimisation of (9) (Anderson & Bahadur, 1962). Thus, an iterative procedure is inevitable in order to obtain the Bayes optimal linear classifier.

In the work of Marks and Dunn (1974), for example, the iterative procedure described is to solve for $\mathbf{w}$ and w_0 as given by

$$\begin{aligned}\mathbf{w} &= \left[s_1\boldsymbol{\Sigma}_1 + s_2\boldsymbol{\Sigma}_2\right]^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\\ w_0 &= \mu_1 - s_1\sigma_1^2 = \mu_2 + s_2\sigma_2^2\end{aligned} \tag{10}$$

by obtaining the optimal values of s_1 and s_2 via systematic trial and error. We denote this heteroscedastic LDA procedure by R-HLD-2, for the reason that the two parameters s_1 and s_2 are chosen at random.

Anderson and Bahadur (1962) make the observation that if the weight vector $\mathbf{w}$ and the threshold w_0 are both multiplied by the same positive scalar, the decision boundary remains unchanged. Therefore, by multiplying (10) through by the scalar $s_1 + s_2$, $\mathbf{w}$ and w_0 can be put in the form of:

$$\begin{aligned}\mathbf{w} &= \left[s\boldsymbol{\Sigma}_2 + (1-s)\boldsymbol{\Sigma}_1\right]^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\\ w_0 &= \mu_1 - (1-s)\sigma_1^2 = \mu_2 + s\sigma_2^2\end{aligned} \tag{11}$$

Still, the optimal value of s has to be chosen by systematic trial and error. We denote this heteroscedastic LDA approach by R-HLD-1, for the reason that only one parameter s is chosen at random. As we show in the next section, s is unbounded. Therefore, the difficulty faced by this approach is that s has to be chosen from the interval $(-\infty, \infty)$, so that the probability of finding the optimal s for a given dataset is low, without extensive trial and error to limit the choice of s to some finite interval $[a, b]$.

To avoid the unguided trial and error procedure in Anderson and Bahadur (1962); Marks and Dunn (1974), (Peterson & Mattson, 1966) and Fukunaga (2013, Chapter 4) propose a theoretical approach described below:

1. Change s from 0 to 1 with small step increments Δs.
2. Evaluate $\mathbf{w}$ as given by:

$$\mathbf{w} = \left[s\boldsymbol{\Sigma}_1 + (1-s)\boldsymbol{\Sigma}_2\right]^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{12}$$

3. Evaluate w_0 as given by:

$$w_0 = \frac{s\mu_2\sigma_1^2 + (1-s)\mu_1\sigma_2^2}{s\sigma_1^2 + (1-s)\sigma_2^2} \tag{13}$$

4. Compute the probability of misclassification p_e.
5. Choose $\mathbf{w}$ and w_0 that minimise p_e.

We refer to this procedure as C-HLD, for the reason that the optimal s is constrained in the interval [0, 1].

However, we highlight two main problems with the above C-HLD procedure:

1. There is no obvious choice of the step rate Δs. Too small a value of Δs will demand too many matrix inversions in Step 2, as there will be too many s values. On the other hand, if Δs is too large, the optimal s may not be refined enough, and the $\mathbf{w}$ obtained may not be optimal. Specifically, the change in $\mathbf{w}$ that results from a small change in s is given as:

$$\begin{aligned}\mathrm{d}\mathbf{w} = &\left(s\boldsymbol{\Sigma}_2 + (1-s)\boldsymbol{\Sigma}_1\right)^{-1}(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2)\\ &\times\left(s\boldsymbol{\Sigma}_2 + (1-s)\boldsymbol{\Sigma}_1\right)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\mathrm{d}s\end{aligned} \tag{14}$$

   which can affect the classification accuracy.
2. The solution obtained this way is only locally optimal as s is bounded in the interval [0, 1]. As we show in the next section, s is actually unbounded. When there is a class imbalance (Xue & Titterington, 2008), the optimal s may be found outside the interval [0, 1] which can lead to poor classification accuracy.

Our proposed algorithm, which is described in the next section, unlike the trial and error approach by Anderson and Bahadur (1962); Marks and Dunn (1974), has a principled optimisation procedure, and unlike Fukunaga (2013); Peterson and Mattson (1966) do not encounter the problem of choosing an inappropriate Δs, nor restricts s to the interval [0, 1]. Consequently, our proposed algorithm achieves a far lower training time than the C-HLD, R-HLD-1 and R-HLD-2, for roughly the same classification accuracy.

3. Gaussian linear discriminant

Let $\mathbf{w} \in \mathbb{R}^d$ be a vector of weights, and $w_0 \in \mathbb{R}$, a threshold such that:

$$\mathcal{C}^*(\mathbf{x}) = \begin{cases}\mathcal{C}_1 & \text{if } \ y = \mathbf{w}^T\mathbf{x} \geq w_0\\ \mathcal{C}_2 & \text{if } \ y = \mathbf{w}^T\mathbf{x} < w_0\end{cases} \tag{15}$$

Since $\mathbf{x}$ is assumed to have a multivariate normal distribution in classes $\mathcal{C}_1$ and $\mathcal{C}_2$, y has a mean of μ_1 and a variance of σ_1^2 for class $\mathcal{C}_1$ and a mean of μ_2 and a variance of σ_2^2 for class $\mathcal{C}_2$ given as:

$$\mu_1 = \mathbf{w}^T\bar{\mathbf{x}}_1 \quad \mu_2 = \mathbf{w}^T\bar{\mathbf{x}}_2 \quad \sigma_1^2 = \mathbf{w}^T\boldsymbol{\Sigma}_1\mathbf{w} \quad \sigma_2^2 = \mathbf{w}^T\boldsymbol{\Sigma}_2\mathbf{w} \tag{16}$$

With reference to the Bayes error of (9), the individual misclassification probabilities can be expressed as:

$$p(y < w_0|\mathcal{C}_1) = \int_{-\infty}^{w_0} \frac{1}{\sqrt{2\pi}\sigma_1}\exp\left[-\frac{(\zeta-\mu_1)^2}{2\sigma_1^2}\right]d\zeta = 1 - Q\left(\frac{w_0-\mu_1}{\sigma_1}\right) \tag{17}$$

and

$$p(y \geq w_0|\mathcal{C}_2) = \int_{w_0}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_2}\exp\left[-\frac{(\zeta-\mu_2)^2}{2\sigma_2^2}\right]d\zeta = Q\left(\frac{w_0-\mu_2}{\sigma_2}\right) \tag{18}$$

where $Q(.)$ is the Q-function. Therefore, the Bayes error to be minimised may be rewritten as:

$$p_e = \pi_1\left[1 - Q(z_1)\right] + \pi_2\left[Q(z_2)\right] \tag{19}$$

where

$$z_1 = \frac{w_0-\mu_1}{\sigma_1} \quad \text{and} \quad z_2 = \frac{w_0-\mu_2}{\sigma_2} \tag{20}$$

Our aim is to find a local minimum of p_e. A necessary condition is for the gradient of p_e to be zero, i.e.,

$$\nabla p_e(\mathbf{w}, w_0) = \left[\frac{\partial p_e}{\partial \mathbf{w}^T}, \frac{\partial p_e}{\partial w_0}\right]^T = \mathbf{0} \tag{21}$$

From (9), it can be shown that:

$$\frac{\partial p_e}{\partial \mathbf{w}} = \pi_1\left(\frac{1}{\sqrt{2\pi}}e^{-z_1^2/2}\frac{\partial z_1}{\partial \mathbf{w}}\right) - \pi_2\left(\frac{1}{\sqrt{2\pi}}e^{-z_2^2/2}\frac{\partial z_2}{\partial \mathbf{w}}\right) \tag{22}$$

From (20), however, we obtain the following:

$$\frac{\partial z_1}{\partial \mathbf{w}} = \frac{-\sigma_1\bar{\mathbf{x}}_1 - z_1\boldsymbol{\Sigma}_1\mathbf{w}}{\sigma_1^2} \quad \text{and} \quad \frac{\partial z_2}{\partial \mathbf{w}} = \frac{-\sigma_2\bar{\mathbf{x}}_2 - z_2\boldsymbol{\Sigma}_2\mathbf{w}}{\sigma_2^2} \tag{23}$$

Therefore,

$$\frac{\partial p_e}{\partial \mathbf{w}} = \frac{1}{\sqrt{2\pi}}\left[-\pi_1 e^{-z_1^2/2}\left(\frac{\sigma_1\bar{\mathbf{x}}_1 + z_1\boldsymbol{\Sigma}_1\mathbf{w}}{\sigma_1^2}\right) + \pi_2 e^{-z_2^2/2}\left(\frac{\sigma_2\bar{\mathbf{x}}_2 + z_2\boldsymbol{\Sigma}_2\mathbf{w}}{\sigma_2^2}\right)\right] \tag{24}$$

It can similarly be shown from (9) that,

$$\frac{\partial p_e}{\partial w_0} = \pi_1\left(\frac{1}{\sqrt{2\pi}}e^{-z_1^2/2}\frac{\partial z_1}{\partial w_0}\right) - \pi_2\left(\frac{1}{\sqrt{2\pi}}e^{-z_2^2/2}\frac{\partial z_2}{\partial w_0}\right) \tag{25}$$

Again, from (20),

$$\frac{\partial z_1}{\partial w_0} = \frac{1}{\sigma_1} \quad \text{and} \quad \frac{\partial z_2}{\partial w_0} = \frac{1}{\sigma_2} \tag{26}$$

Therefore,

$$\frac{\partial p_e}{\partial w_0} = \frac{\pi_1}{\sqrt{2\pi}}\left(\frac{1}{\sigma_1}e^{-z_1^2/2}\right) - \frac{\pi_2}{\sqrt{2\pi}}\left(\frac{1}{\sigma_2}e^{-z_2^2/2}\right) \tag{27}$$

Now, equating the gradient $\nabla p_e(\mathbf{w}, w_0)$ to zero, the following set of equations are obtained:

$$\left(\frac{\pi_2 z_2}{\sigma_2^2}e^{-z_2^2/2}\boldsymbol{\Sigma}_2 - \frac{\pi_1 z_1}{\sigma_1^2}e^{-z_1^2/2}\boldsymbol{\Sigma}_1\right)\mathbf{w} = \left(\frac{\pi_1}{\sigma_1}e^{-z_1^2/2}\right)\bar{\mathbf{x}}_1 - \left(\frac{\pi_2}{\sigma_2}e^{-z_2^2/2}\right)\bar{\mathbf{x}}_2 \tag{28}$$

$$\frac{\pi_1}{\sigma_1}e^{-z_1^2/2} = \frac{\pi_2}{\sigma_2}e^{-z_2^2/2} \tag{29}$$

Substituting (29) into (28) yields:

$$\left(\frac{z_2}{\sigma_2}\boldsymbol{\Sigma}_2 - \frac{z_1}{\sigma_1}\boldsymbol{\Sigma}_1\right)\mathbf{w} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{30}$$

Then the vector $\mathbf{w}$ can be given by:

$$\mathbf{w} = \left(\frac{z_2}{\sigma_2}\boldsymbol{\Sigma}_2 - \frac{z_1}{\sigma_1}\boldsymbol{\Sigma}_1\right)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{31}$$

It will be noted however that (31) is still in terms of w_0, so that an explicit representation of w_0 in terms of $\mathbf{w}$ is needed from (29) to substitute in z_1 and z_2 in (31). This is where our approach most significantly differs from Fukunaga (2013). Solving for w_0 from (29) results in the following quadratic:

$$\frac{z_2^2}{2} - \frac{z_1^2}{2} - \ln\left(\frac{\tau\sigma_1}{\sigma_2}\right) = 0 \tag{32}$$

which can be simplified to:

$$\left(\frac{w_0-\mu_2}{\sigma_2}\right)^2 - \left(\frac{w_0-\mu_1}{\sigma_1}\right)^2 - 2\ln\frac{\tau\sigma_1}{\sigma_2} = 0, \tag{33}$$

where τ is given as before as $\tau = \pi_2/\pi_1$. If τ is defined and not equal to zero, and $\sigma_1^2 \neq \sigma_2^2$ (since $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$ for heteroscedastic LDA), (33) can be shown to have the following solutions:

$$w_0 = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 \pm \sigma_1\sigma_2\sqrt{(\mu_1-\mu_2)^2 + 2(\sigma_1^2-\sigma_2^2)\ln\left(\frac{\tau\sigma_1}{\sigma_2}\right)}}{\sigma_1^2-\sigma_2^2} \tag{34}$$

Nevertheless, since there are two solutions to w_0 in (34), a choice has to be made as to which of them is substituted into (31). To eliminate one of the solutions, we consider the second-order partial derivative of p_e with respect to w_0 evaluated at w_0 as given by (34), and determine under what condition it is greater than or equal to zero. This is a second-order necessary condition for p_e to be a local minimum. From (27), it can be shown that:

$$\frac{\partial^2 p_e}{\partial w_0^2} = \frac{\pi_1}{\sqrt{2\pi}}\left(-\frac{z_1}{\sigma_1^2}e^{-z_1^2/2}\right) + \frac{\pi_2}{\sqrt{2\pi}}\left(\frac{z_2}{\sigma_2^2}e^{-z_2^2/2}\right) \tag{35}$$

We denote this second-order derivative by h. We then consider all possibilities of z_1 and z_2 (which are the variables in (35) that depend on w_0) under three cases, and analyse the sign of h in each.

Case 1

$z_2 \leq 0$ and $z_1 \geq 0$: then h is trivially non-positive.

Case 2

$z_2 \geq 0$ and $z_1 \leq 0$: then h is trivially non-negative.

Case 3

$z_2 > 0$ and $z_1 > 0$ or $z_2 < 0$ and $z_1 < 0$: then h is non-negative if and only if

$$\ln\left(\frac{\pi_2 z_2}{\sigma_2^2}\right) - \frac{z_2^2}{2} \geq \ln\left(\frac{\pi_1 z_1}{\sigma_1^2}\right) - \frac{z_1^2}{2} \tag{36}$$

i.e.,

$$\ln\left(\frac{z_2}{\sigma_2}\Big/\frac{z_1}{\sigma_1}\right) \geq \frac{z_2^2}{2} - \frac{z_1^2}{2} - \ln\left(\frac{\tau\sigma_1}{\sigma_2}\right) \tag{37}$$

It will be noted that the right-hand side of the inequality (37) is identically zero, as can be seen from (32). Therefore, the condition under which h is greater than or equal to zero is when:

$$\frac{z_2}{\sigma_2} \geq \frac{z_1}{\sigma_1} \tag{38}$$

Note also that Case 2 necessarily satisfies (38) so that we consider (38) as the general inequality for the non-negativity of h for all cases, and thus for w_0 to be a local minimum.

Now, when one considers the two solutions of w_0 in (35), only the solution given by:

$$w_0 = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 + \sigma_1\sigma_2\sqrt{(\mu_1-\mu_2)^2 + 2(\sigma_1^2-\sigma_2^2)\ln\left(\frac{\tau\sigma_1}{\sigma_2}\right)}}{\sigma_1^2-\sigma_2^2} \tag{39}$$

satisfies the inequality of (38), i.e., only this choice of w_0 corresponds to a local minimum. The proof of this is given in the appendix.

We may then substitute this expression of w_0 into (31) so that (31) is in terms of $\mathbf{w}$ only. Even so, $\mathbf{w}$ has to be solved for iteratively. This is because (31) has no closed-form solution since μ_1, μ_2, σ_1, σ_2 are themselves functions of $\mathbf{w}$. As the iterative procedure requires an initial choice of $\mathbf{w}$, we use Fisher's choice of the weight vector as given by:

$$\mathbf{w} = (n_1\boldsymbol{\Sigma}_1 + n_2\boldsymbol{\Sigma}_2)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{40}$$

as our initial solution. Again, we mention that n_1 and n_2 are the cardinalities of $\mathcal{D}_1$ and $\mathcal{D}_2$. After a number of such iterative updates, the optimal w_0 is then solved for from (39). This algorithm, known as the Gaussian Linear Discriminant (GLD), is described in detail in Algorithm 1.

Algorithm 1 GLD.

1: Input: $\mathcal{D}_1$ and $\mathcal{D}_2$
2: Evaluate $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$
3: Initialise $\mathbf{w}$: $\mathbf{w} = (n_1\boldsymbol{\Sigma}_1 + n_2\boldsymbol{\Sigma}_2)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$
4: Evaluate $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, z_1, z_2$.
5: **while** Stopping criteria are not satisfied **do**
6: Solve for w_0 from (39)
7: Evaluate z_1, z_2
8: Evaluate the Bayes error p_e
9: Update $\mathbf{w}$ as $\mathbf{w} = \left(\frac{z_2}{\sigma_2}\boldsymbol{\Sigma}_2 - \frac{z_1}{\sigma_1}\boldsymbol{\Sigma}_1\right)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$
10: Evaluate $\mu_1, \mu_2, \sigma_1, \sigma_2$.
11: **end while**

Note that by multiplying both $\mathbf{w}$ of (31) and w_0 proportionally by $c = (\sigma_1 z_2 - \sigma_2 z_1)/\sigma_1\sigma_2$ (due to (38), c is non-negative and hence the discrimination criterion given by (15) is not changed), the GLD may be viewed in terms of the optimal solution of (11), where

$$s = -\sigma_2 z_1/(\sigma_1 z_2 - \sigma_2 z_1). \tag{41}$$

which is unbounded given the inequality of (38). However, unlike Anderson and Bahadur (1962); Marks and Dunn (1974), s is not chosen by systematic trial and error, and unlike Fukunaga (2013), s is not varied between 0 and 1 at small step increments. Instead, since s is a function of $\mathbf{w}$ and w_0, our algorithm may be interpreted as obtaining increasingly refined values of s by improving upon $\mathbf{w}$ and w_0 starting from Fisher's solution, as is described in Algorithm 1.

3.1. Stopping criteria

The GLD algorithm may be terminated under any of the following conditions:

1. When the change in the objective function p_e remains within a certain tolerance ϵ_1 for a number of consecutive iterations.
2. When the change in the norm of $\mathbf{w}$ remains within a certain tolerance ϵ_2 for a number of consecutive iterations.
3. When the gradient of p_e as given by (21) remains within a certain tolerance ϵ_3 for a number of consecutive iterations.
4. After a fixed number of iterations I, if convergence is slow.

At the end of the algorithm, the final solution may be chosen either as the solution to which the iterations converge, or the solution corresponding to the minimum p_e found in the iterative updates.

3.2. Multiclass classification

Suppose now that there are $K > 2$ classes in the dataset $\mathcal{D}$, then the classification problem may be reduced to a number of binary classification problems. The two main approaches usually taken for this reduction are the One-vs-All (OvA) and One-vs-One (OvO) strategies (Bishop, 2006; Hsu & Lin, 2002).

3.2.1. One-vs-All (OvA)

In OvA, one trains a classifier to discriminate between one class and all other classes. Thus, there are K different classifiers. An unknown vector $\mathbf{x}$ is then tested on all K classifiers so that the class corresponding to the classifier with the highest discriminant score is chosen. However, with respect to the proposed GLD algorithm, this is an ill-suited approach. This is because the collection of all other classes on one side of the discriminant will not necessarily have a normal distribution, and could in fact be multimodal, if the means are well-separated. Since our algorithm is built on strong normality assumptions of the data on each side of the discriminant, the GLD, as has been formulated, is expected to perform poorly.

3.2.2. One-vs-One

In OvO, a classifier is trained to discriminate between every pair of classes in the dataset, ignoring the other $K-2$ classes. Thus, there are $K(K-1)/2$ unique classifiers that may be constructed. Again, an unknown vector $\mathbf{x}$ is tested on all $K(K-1)/2$ classifiers. The predicted classes for all the classifiers are then tallied so that the class that occurs most frequently is chosen. This is equivalent to a majority vote decision. In a lot of cases, however, there is no clear-cut winner, as more than one class may have the highest number of votes. In such a case, the most likely class is often chosen randomly between those most frequently occurring classes. The GLD provides a more appropriate means for breaking such ties, by making use of the minimised Bayes error p_e for each classifier. Specifically, one may instead use a weighted voting system, where the count of every predicted class is weighted by $1 - p_e$, since p_e provides an appropriate measure of uncertainty associated with each classifier output. Thus, the decision rule is reduced to choosing the maximum weighted vote among the K classes.

Note that even though the GLD minimises the Bayes error for each classifier, the overall Bayes error for a multiclass problem may not be minimised by using multiple binary classifiers.

4. Non-normal distributions

So far, the fundamental assumption that has been used to derive the GLD is that the data in each class has a normal distribution. Thus, for an unknown non-normal distribution, the linear classifier we have obtained does not minimise the Bayes error for that unknown distribution. We argue, however, that if this unknown distribution is nearly-normal (Mudholkar & Hutson, 2000), then a more robust linear classifier may be found in some neighbourhood of the GLD. For this reason, we use a local neighbourhood search algorithm to explore the region in $\mathbb{R}^{d+1}$ around the GLD to obtain the classifier that minimises the number of misclassifications on the training dataset. We do this by perturbing each of the $d+1$ vector elements in the optimal $\tilde{\mathbf{w}} = [w_0, \mathbf{w}^T]^T$ obtained from the GLD procedure by a small amount $\delta\tilde{w}_i$. After every perturbation, the resulting classifier is evaluated on the test dataset.

Table 1
List and characteristics of datasets.

Dataset	Label	n	d	K
D1	(a)	2000	8	2
D2	(b)	2000	4	2
Liver	(c)	345	6	2
Shuttle	(d)	58,000	9	7
Vowels	(e)	990	10	11
Zernike Moments	(f)	2000	47	10
Image Segmentation (Statlog)	(g)	2310	19	7
Spambase	(h)	4601	37	2
Wine Quality (White)	(i)	4898	11	7
Pen Digits	(j)	5620	64	10
Satellite (Statlog)	(k)	6435	36	6
Letters	(l)	20,000	16	26

This table lists the datasets used in the experimental section. K is the number of classes, d is the dimensionality of the dataset, and n is the number of data points in the dataset.

Table 2
Average Bayes error (%).

Dataset	LDA	C-HLD	R-HLD-1	R-HLD-2	GLD
(a)	0.0397	0.0382	0.0383	0.0361	**0.0360**
(b)	0.0774	0.0749	0.0749	0.0740	**0.0739**
(c)	0.9981	**0.9838**	**0.9838**	**0.9838**	**0.9838**
(d)	**0.0001**	**0.0001**	**0.0001**	**0.0001**	**0.0001**
(e)	0.0339	**0.0326**	**0.0326**	**0.0326**	**0.0326**
(f)	0.0054	0.0051	**0.0048**	**0.0048**	0.0050
(g)	0.0037	**0.0029**	**0.0029**	**0.0029**	**0.0029**
(h)	0.0253	**0.0228**	**0.0228**	**0.0228**	**0.0228**
(i)	0.0162	0.0201	0.0156	0.0155	**0.0154**
(j)	**0.0002**	**0.0002**	**0.0002**	**0.0002**	**0.0002**
(k)	0.0046	**0.0039**	**0.0039**	**0.0039**	**0.0039**
(l)	**0.0007**	**0.0007**	**0.0007**	**0.0007**	**0.0007**

This table shows the average Bayes error per discriminant as a percentage for each dataset for LDA, GLD, C-HLD, R-HLD-1 and R-HLD-2. Best values are in bold.

This procedure is repeated as described in Algorithm 2 until the

Algorithm 2 Local neighbourhood search (LNS).

Input: Optimal $\tilde{\mathbf{w}} = [w_0, \mathbf{w}^T]^T$ obtained from the GLD.
while Stopping criterion is not satisfied **do**
  Let $\tilde{\mathbf{w}}$ be the current solution.
  for $i \leftarrow 1$ to d **do**
    $\mathbf{v}^+ \leftarrow \tilde{\mathbf{w}}$, $\mathbf{v}^- \leftarrow \tilde{\mathbf{w}}$.
    $\mathbf{v}^+ \leftarrow v_i^+ + \delta v_i^+$
    Evaluate the misclassifications on the training set using $\mathbf{v}^+$
    $\mathbf{v}^- \leftarrow v_i^- - \delta v_i^-$
    Evaluate the misclassifications on the training set using $\mathbf{v}^-$
  end for
  Set the classifier with the minimum number of misclassifications as the current solution $\tilde{\mathbf{w}}$.
end while
Choose the classifier with the smallest number of misclassifications.

stopping criterion is satisfied.

The algorithm is terminated after a certain maximum number of iterations R is reached. Additionally, one may perform an early termination if after a predefined number of iterations $r_{\max}$, there is no improvement in the minimum number of misclassifications on the training dataset that has been found in the search.

5. Experimental validation

We validate our proposed algorithm on two artificial datasets denoted by D1 and D2, as well as on ten real-world datasets taken from the University of California, Irvine (UCI) Machine Learning Repository. These datasets are shown in Table 1, and cut across a wide range of applications including handwriting recognition, medical diagnosis, remote sensing and spam filtering. D1 and D2 are normally distributed with different covariance matrices. For D1, we generate 1000 samples for class $\mathcal{C}_1$ and 2000 samples for class $\mathcal{C}_2$ using the following Gaussian parameters:

$$\begin{aligned}\bar{\mathbf{x}}_2 &= [3.86, 3.10, 0.84, 0.84, 1.64, 1.08, 0.26, 0.01]^T,\\ \mathbf{\Sigma}_2 &= \mathrm{diag}(8.41, 12.06, 0.12, 0.22, 1.49, 1.77, 0.35, 2.73)\\ \bar{\mathbf{x}}_1 &= \bar{\mathbf{x}}_2 - 0.3, \quad \mathbf{\Sigma}_1 = \mathbf{I}\end{aligned} \tag{42}$$

For D2, we generate 2000 samples for class $\mathcal{C}_1$ and 4000 samples for class $\mathcal{C}_2$ using the following Gaussian parameters:

$$\bar{\mathbf{x}}_2 = [-1.5, -0.75, 0.75, 1.5]^T,$$

Table 3
Average classification accuracy (%).

Dataset	LDA	C-HLD	R-HLD-1	R-HLD-2	GLD	LNS	SVM
(a)	76.00	77.18	77.00	78.48	**78.65**	78.57	77.47
(b)	76.87	77.93	77.93	78.17	**78.37**	78.00	77.70
(c)	67.83	63.19	62.32	62.03	63.77	68.12	**68.70**
(d)	94.10	96.60	96.74	96.73	96.59	**97.91**	84.39
(e)	73.64	74.14	74.44	74.44	74.14	75.66	**76.77**
(f)	84.00	83.90	84.10	84.15	**84.80**	84.00	81.90
(g)	94.33	94.59	94.59	94.63	94.59	94.89	**96.15**
(h)	88.76	88.29	88.26	88.15	88.26	**90.28**	85.68
(i)	53.41	46.59	53.37	53.33	53.55	**54.14**	51.88
(j)	96.74	96.99	96.97	96.98	97.01	97.41	**97.84**
(k)	85.69	86.06	86.06	86.03	86.08	86.65	**86.85**
(l)	81.67	81.87	81.83	81.78	81.88	82.25	**85.39**

This table shows the average classification accuracy (%) on the test datasets for LDA, C-HLD, R-HLD-1, R-HLD-2, GLD, GLD+LNS and SVM. Best values are in bold.

$$\begin{aligned}\mathbf{\Sigma}_2 &= \mathrm{diag}(0.25, 0.75, 1.25, 1.75)\\ \bar{\mathbf{x}}_1 &= \bar{\mathbf{x}}_2 - 0.75, \quad \mathbf{\Sigma}_1 = \mathbf{I}\end{aligned} \tag{43}$$

The above Gaussian parameters are slightly modified from the two class data used by Fukunaga (2013) and Xue and Titterington (2008) in order to make the sample means less separated.

For each dataset in Table 1, we perform 10-fold cross validation. We run 20 different trials. On each training dataset, we evaluate the minimum Bayes error achievable by our proposed algorithm averaged over all 10 folds and 20 trials. If there are more than two classes, we use OvO, and calculate the mean Bayes error over all $K(K-1)/2$ discriminants. As we are interested only in linear classification, we compare the performance of the GLD with the original LDA as well as the heteroscedastic LDA procedures by Fukunaga (2013),Anderson and Bahadur (1962) and Marks and Dunn (1974) as described in Section 2 in terms of the Bayes error (9). For the sake of brevity, we denote these three heteroscedastic LDA algorithms by the annotations earlier introduced: C-HLD, R-HLD-1 and R-HLD-2 respectively. These results are shown in Table 2.

Moreover, for each of the test datasets, we evaluate the average classification accuracy for each of LDA, C-HLD, R-HLD-1, R-HLD-2, GLD and GLD with local neighbourhood search (LNS). We also compare the performance of these LDA approaches to the SVM. These results are shown in Table 3, while the average training times of the algorithms are shown in Table 4.

We estimate the prior probabilities based on the relative frequencies of the data in each class in the dataset, and the stopping criterion for the GLD is thus: we stop if the gradient of $\mathbf{w}$ change is less than or equal to $\epsilon_3 = 10^{-6}$, or else we terminate our algorithm after $I = 20$ iterations and choose the solution corresponding to the minimum p_e. Also, for the LNS procedure, we perturb each vector

Table 4
Average training time (s).

Dataset	LDA	C-HLD	R-HLD-1	R-HLD-2	GLD	LNS	SVM
(a)	**0.001**	0.161	0.140	0.139	0.002	0.181	23.192
(b)	**0.001**	0.142	0.121	0.121	0.002	0.060	0.721
(c)	**0.001**	0.155	0.1415	0.1337	0.003	0.028	2.673
(d)	**0.037**	3.531	3.023	3.012	0.089	43.32	4623.138
(e)	**0.036**	11.099	9.409	9.751	0.167	2.075	1.173
(f)	**0.387**	123.662	123.649	121.906	1.955	110.694	23.126
(g)	**0.128**	37.320	30.876	37.875	0.488	2.143	21.775
(h)	**0.101**	10.437	7.729	7.474	0.753	36.83	804.574
(i)	**0.017**	4.257	3.691	3.750	0.080	5.928	914.257
(j)	**0.638**	10.099	9.358	9.171	0.915	168.19	409.38
(k)	**0.304**	18.067	17.842	17.912	0.858	13.919	311.202
(l)	**0.835**	73.050	64.022	65.414	3.245	37.202	109.232

This table shows the average training times on the test datasets for LDA, C-HLD, R-HLD-1, R-HLD-2, GLD, GLD+LNS and SVM. Best values are in bold.

element by 10% of its absolute value, i.e. $\delta\tilde{w}_i = 0.1|\tilde{w}_i|$, and we run for R=1000 iterations, terminating prematurely if $r_{\max} = 0.1R$. We use a step size of $\Delta s = 0.001$ for the C-HLD algorithm, and run 1000 trials for R-HLD-1 and R-HLD-2. All the parameters used in the experiments are optimised via cross-validation. Note that if the sample covariance matrix is singular, we use the Moore–Penrose pseudo-inverse.

6. Results and discussion

For real-world datasets, the covariance matrices of the classes are rarely equal, therefore the homoscedasticity assumption in LDA does not hold. Our results in Table 2 confirm that LDA does not minimise the Bayes error under heteroscedasticity, as none of the datasets used has equal covariance matrices. With the exception of datasets (d), (j) and (l), where LDA achieves an equal Bayes error as the other heteroscedastic LDA approaches, LDA is outperformed by the GLD on all remaining datasets in terms of minimising the Bayes error. It will be noted that the other three heteroscedastic LDA approaches algorithms achieve a performance comparable to the GLD on all the datasets in terms of the Bayes error. However, R-HLD-1 and R-HLD-2 require a lot of trials (1000 in our experiments) in order to obtain the optimal parameters s and s_1, s_2 respectively, while C-HLD requires a step size of $\Delta s = 0.001$ which translates to 1001 trials. Consequently, the training time for these algorithms far exceed that of the GLD, as can be seen in Table 4. For example, the gain in training time of the GLD over C-HLD, R-HLD-1 and R-HLD-2 is over 62 folds for dataset (g), and about 20 folds for dataset (l). Moreover, since C-HLD, R-HLD-1 and R-HLD-2 all require matrix inversions, performing a matrix inversion for each of the 1000 trials can be a computationally demanding task especially for high-dimensional data, which have large covariance matrices. Instead, since the GLD follows a principled optimisation procedure, the number of matrix inversions required is far lower. For example, on dataset (f), which has a dimensionality of 47, the GLD requires over 60 times less time to train than the other heteroscedastic LDA approaches.

It is conceivable that the minimisation of the Bayes error would translate into a good performance in terms of the classification accuracy, if the normality assumption of LDA holds. For this reason, it can be seen in Table 3 that the GLD achieves the best classification accuracy on datasets (a) and (b), which are generated from known normal distributions. Thus, the proposed GLD algorithm is particularly suited for applications with datasets that tend to be normally distributed in each class e.g. in machine fault diagnosis, or accelerometer-based human activity recognition (Ojetola, Gaura, & Brusey, 2015), as it also requires far less training time than the existing heteroscedastic LDA approaches.

However, for datasets (c) through to (l), the classes do not have any known normal distribution. Therefore, minimising the Bayes error under the normality assumption would not necessarily result in a classifier that has the best classification accuracy, even if the difference in covariance matrices has been accounted for. For this reason, it is not surprising that LDA achieves a superior classification accuracy than C-HLD, R-HLD-1, R-HLD-2 and the GLD on datasets (c) and (h) as can be seen in Table 3. However, by searching around the neighbourhood of the GLD, the local neighbourhood search (LNS) algorithm is able to account for the non-normality and obtain a more robust classifier. Thus, the GLD, together with the LNS procedure, achieves a higher classification accuracy than all the LDA approaches on all the real-world datasets (i.e. (c)–(l)) with the exception of dataset (f) which has the GLD showing superior classification accuracy.

While the SVM outperforms the LDA approaches on half of the datasets, its training time can be rather long for large datasets. For instance, for dataset (d) which has 58000 elements, the SVM takes about 1.3 h to train whereas the GLD with LNS, which achieves the best classification accuracy on this dataset, takes 43 s to train, representing over 100 fold savings in computational time over the SVM. Similar patterns can be seen in other datasets like (i), where the GLD with LNS achieves a superior classification accuracy with over 150 times shorter training time than the SVM. This suggests that for such large datasets, the GLD with local neighbourhood search is a low-complexity alternative to the SVM, as it requires far less computational time than the SVM.

We, however, make note of two weaknesses our proposed algorithms have. For the GLD, the procedure as described in Algorithm 1, may converge to a saddle point, instead of a local minimum. Even if it were to converge to a local minimum, there is no guarantee that is the global optimum solution due to the fact that the objective function p_e is known to be non-convex (Anderson & Bahadur, 1962). Also, since the local neighbourhood search involves evaluating the misclassification rate on the training set for every perturbation, the procedure does not scale well with large amounts of training data. Because of this, it is important to have a good initial solution like the GLD, so that an early termination may be performed if there is no improvement after some number of iterations.

7. Conclusion

In this paper, we have presented the Gaussian Linear Discriminant (GLD), a novel and computationally efficient method for obtaining a linear discriminant for heteroscedastic Linear Discriminant Analysis (LDA) for the purpose of binary classification. Our algorithm minimises the Bayes error via an iterative optimisation procedure that uses Fisher's Linear Discriminant as the initial so-

lution. Moreover, the GLD does not require any parameter adjustments. We have also proposed a local neighbourhood search method by which a more robust linear classifier may be obtained for non-normal distributions. Our experimental results on two artificial and ten real world applications show that when the covariance matrices of the classes are unequal, LDA is unable to minimise the Bayes error. Thus, under heteroscedasticity, our proposed algorithm achieves superior classification accuracy to the LDA for normally distributed classes. While the proposed GLD algorithm compares favourably with other heteroscedastic LDA approaches, the GLD requires a far less training time. Moreover, the GLD, together with the LNS, has been shown to be particularly robust, comparing favourably with the SVM, but requiring far less training time on our datasets. Thus, for expert systems like machine fault diagnosis or human activity monitoring that require linear classification, the proposed algorithms provide a low-complexity, high-accuracy solution.

While this work has focused on linear classification, on-going work is focused on modifying the GLD procedure for the purpose of linear dimensionality reduction. Moreover, it is of particular interest to us to be able to derive the Bayes error for some known non-normal distributions. An alternative to this is to be able to obtain a kernel that implicitly transforms some data of a known non-normal distribution into a feature space where the classes are normally distributed. Finally, like all local search algorithms, the performance and complexity of the LNS procedure depends on the choice of the initial solution. Therefore, further work that explores the use initial solutions (including the heteroscedastic LDA approaches discussed) other than the GLD for the LNS procedure is being done.

Appendix A

Theorem 1. *Let w_0^+ and w_0^- be the two distinct solutions of (34), then w_0^+ and w_0^- cannot both satisfy (38) given that $\sigma_1 \neq \sigma_2$.*

Proof. Let

$$\beta = \sqrt{(\mu_1 - \mu_2)^2 + 2(\sigma_1^2 - \sigma_2^2)\ln\left(\frac{\tau\sigma_1}{\sigma_2}\right)} \tag{A.1}$$

and let

$$w_0^+ = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 + \sigma_1\sigma_2\beta}{\sigma_1^2 - \sigma_2^2} \tag{A.2}$$

Then

$$\frac{z_2}{\sigma_2} = \frac{(\mu_2 - \mu_1)\sigma_2 + \beta\sigma_1}{\sigma_2(\sigma_1^2 - \sigma_2^2)}, \quad \frac{z_1}{\sigma_1} = \frac{(\mu_2 - \mu_1)\sigma_1 + \beta\sigma_2}{\sigma_1(\sigma_1^2 - \sigma_2^2)} \tag{A.3}$$

Suppose that w_0^+ satisfies (38), then

$$\frac{(\mu_2 - \mu_1)\sigma_2 + \beta\sigma_1}{\sigma_2(\sigma_1^2 - \sigma_2^2)} \geq \frac{(\mu_2 - \mu_1)\sigma_1 + \beta\sigma_2}{\sigma_1(\sigma_1^2 - \sigma_2^2)} \tag{A.4}$$

i.e.,

$$\frac{\beta\sigma_1^2}{\sigma_1^2 - \sigma_2^2} \geq \frac{\beta\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \tag{A.5}$$

This implies that $\sigma_1^2/(\sigma_1^2 - \sigma_2^2) > \sigma_2^2/(\sigma_1^2 - \sigma_2^2)$ since β is a positive scalar.

Consider now w_0^- given as:

$$w_0^- = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 - \sigma_1\sigma_2\beta}{\sigma_1^2 - \sigma_2^2} \tag{A.6}$$

Then

$$\frac{z_2}{\sigma_2} = \frac{(\mu_2 - \mu_1)\sigma_2 - \beta\sigma_1}{\sigma_2(\sigma_1^2 - \sigma_2^2)}, \quad \frac{z_1}{\sigma_1} = \frac{(\mu_2 - \mu_1)\sigma_1 - \beta\sigma_2}{\sigma_1(\sigma_1^2 - \sigma_2^2)} \tag{A.7}$$

In order for (38) to be satisfied, it can be shown, similar to (A.5), that

$$\frac{-\beta\sigma_1^2}{\sigma_1^2 - \sigma_2^2} \geq \frac{-\beta\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \tag{A.8}$$

which can be simplified to give $1 \leq 0$. Since this conclusion is false, only w_0^+ satisfies (26). □

References

Anderson, T. W., & Bahadur, R. (1962). Classification into two multivariate normal distributions with different covariance matrices. *The Annals of Mathematical Statistics*, 420–431.

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

Bishop, C. M. (2006). Pattern recognition. *Machine Learning, 128*, 1–58.

Buturovic, L. J. (1994). Toward Bayes-optimal linear dimension reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*(10), 420–424.

Cai, D., He, X., Zhou, K., Han, J., & Bao, H. (2007). Locality sensitive discriminant analysis. In *Proceedings of the 20th international joint conference on artificial intelligence* (pp. 708–713). Morgan Kaufmann Publishers Inc.

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.

Coomans, D., Jonckheer, M., Massart, D. L., Broeckaert, I., & Blockx, P. (1978). The application of linear discriminant analysis in the diagnosis of thyroid diseases. *Analytica Chimica Acta, 103*(4), 409–415.

Decell, H. P., & Mayekar, S. M. (1977). Feature combinations and the divergence criterion. *Computers & Mathematics with Applications, 3*(1), 71–76.

Decell Jr, H. P., & Marani, S. K. (1976). Feature combinations and the Bhattacharyya criterion. *Communications in Statistics-Theory and Methods, 5*(12), 1143–1152.

Duin, R., & Loog, M. (2004). Linear dimensionality reduction via a heteroscedastic extension of LDA: The Chernoff criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(6), 732–739.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*(2), 179–188.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association, 84*(405), 165–175.

Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. Academic press.

Fukunaga, K., & Mantock, J. (1983). Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 671–678.

Hamsici, O. C., & Martinez, A. M. (2008). Bayes optimality in linear discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(4), 647–657.

Hariharan, B., Malik, J., & Ramanan, D. (2012). Discriminative decorrelation for clustering and classification. In *European conference on computer vision* (pp. 459–472). Springer.

Hastie, T., & Tibshirani, R. (1996). Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 155–176.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks, 13*(2), 415–425.

Izenman, A. J. (2009). *Modern multivariate statistical techniques: Regression, classification, and manifold learning*. Springer Science & Business Media.

Ju, J., Kolaczyk, E. D., & Gopal, S. (2003). Gaussian mixture discriminant analysis and sub-pixel land cover characterization in remote sensing. *Remote Sensing of Environment, 84*(4), 550–560.

Li, Z., Lin, D., & Tang, X. (2009). Nonparametric discriminant analysis for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(4), 755–761.

Liu, J., Chen, S., Tan, X., & Zhang, D. (2007). Efficient pseudoinverse linear discriminant analysis and its nonlinear form for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence, 21*(08), 1265–1278.

Loog, M., & Duin, R. P. (2002). Non-iterative heteroscedastic linear dimension reduction for two-class data. In *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)* (pp. 508–517). Springer.

Lu, J., Plataniotis, K. N., & Venetsanopoulos, A. N. (2003). Regularized discriminant analysis for the small sample size problem in face recognition. *Pattern Recognition Letters, 24*(16), 3079–3087.

Mahmoudi, N., & Duman, E. (2015). Detecting credit card fraud by modified Fisher discriminant analysis. *Expert Systems with Applications, 42*(5), 2510–2516.

Malina, W. (1981). On an extended Fisher criterion for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5), 611–614.

Marks, S., & Dunn, O. J. (1974). Discriminant functions when covariance matrices are unequal. *Journal of the American Statistical Association, 69*(346), 555–559.

McLachlan, G. (2004). *Discriminant analysis and statistical pattern recognition*: 544. John Wiley & Sons.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., & Mullers, K.-R. (1999). Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.* (pp. 41–48). IEEE.

Mudholkar, G. S., & Hutson, A. D. (2000). The epsilon–skew–normal distribution for analyzing near-normal data. *Journal of Statistical Planning and Inference, 83*(2), 291–309.

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems, 2*, 841–848.

Ojetola, O., Gaura, E., & Brusey, J. (2015). Data set for fall events and daily activities from inertial sensors. In *Proceedings of the 6th ACM multimedia systems conference* (pp. 243–248). ACM.

Paliwal, K. K., & Sharma, A. (2012). Improved pseudoinverse linear discriminant analysis method for dimensionality reduction. *International Journal of Pattern Recognition and Artificial Intelligence, 26*(01), 1250002.

Peterson, D., & Mattson, R. (1966). A method of finding linear discriminant functions for a class of performance criteria. *IEEE Transactions on Information Theory, 12*(3), 380–387.

Polat, K., Güneş, S., & Arslan, A. (2008). A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine. *Expert Systems with Applications, 34*(1), 482–487.

Sengur, A. (2008). An expert system based on linear discriminant analysis and adaptive neuro-fuzzy inference system to diagnosis heart valve diseases. *Expert Systems with Applications, 35*(1), 214–222.

Sharma, A., & Paliwal, K. K. (2008). Cancer classification by gradient LDA technique using microarray gene expression data. *Data & Knowledge Engineering, 66*(2), 338–347.

Sharma, A., & Paliwal, K. K. (2015). Linear discriminant analysis for the small sample size problem: An overview. *International Journal of Machine Learning and Cybernetics, 6*(3), 443–454.

Song, F., Zhang, D., Wang, J., Liu, H., & Tao, Q. (2007). A parameterized direct LDA and its application to face recognition. *Neurocomputing, 71*(1), 191–196.

Xue, J.-H., & Titterington, D. M. (2008). Do unbalanced data have a negative effect on LDA? *Pattern Recognition, 41*(5), 1558–1571.

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional datawith application to face recognition. *Pattern Recognition, 34*(10), 2067–2070.

Yuan, G.-X., Ho, C.-H., & Lin, C.-J. (2012). Recent advances of large-scale linear classification. *Proceedings of the IEEE, 100*(9), 2584–2603.

Zhang, W.-Q., & Liu, J. (2008). An equalized heteroscedastic linear discriminant analysis algorithm. *IEEE Signal Processing Letters, 15*, 585–588.

Zhao, Z., Sun, L., Yu, S., Liu, H., & Ye, J. (2009). Multiclass probabilistic kernel discriminant analysis. In *Proceedings of the 21st international jont conference on artificial intelligence* (pp. 1363–1368). Morgan Kaufmann Publishers Inc.