IEEE Open Journal of Signal Processing

Received 24 April 2022; revised 6 July 2022; accepted 14 July 2022. Date of publication 29 July 2022; date of current version 15 August 2022. The review of this article was arranged by Associate Editor Xun Chen.

Digital Object Identifier 10.1109/OJSP.2022.3195150

Weight Vector Tuning and Asymptotic Analysis of Binary Linear Classifiers

LAMA B. NIYAZI [1], ABLA KAMMOUN [1] (Member, IEEE), HAYSSAM DAHROUJ [2] (Senior Member, IEEE), MOHAMED-SLIM ALOUINI [1] (Fellow, IEEE), AND TAREQ Y. AL-NAFFOURI [1] (Senior Member, IEEE)

[1]Electrical and Computer Engineering Program, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia
[2]Center of Excellence for NEOM Research, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia

CORRESPONDING AUTHOR: LAMA B. NIYAZI (email: lama.niyazi@kaust.edu.sa)

This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research under Award OSR-CRG2019-4041.
An extended version of this paper is available on arXiv [1].

ABSTRACT Unlike its intercept, a linear classifier's weight vector cannot be tuned by a simple grid search. Hence, this paper proposes weight vector tuning of a generic binary linear classifier through the parameterization of a decomposition of the discriminant by a scalar which controls the trade-off between conflicting informative and noisy terms. By varying this parameter, the original weight vector is modified in a meaningful way. Applying this method to a number of linear classifiers under a variety of data dimensionality and sample size settings reveals that the classification performance loss due to non-optimal native hyperparameters can be compensated for by weight vector tuning. This yields computational savings as the proposed tuning method reduces to tuning a scalar compared to tuning the native hyperparameter, which may involve repeated weight vector generation along with its burden of optimization, dimensionality reduction, etc., depending on the classifier. It is also found that weight vector tuning significantly improves the performance of Linear Discriminant Analysis (LDA) under high estimation noise. Proceeding from this second finding, an asymptotic study of the misclassification probability of the parameterized LDA classifier in the growth regime where the data dimensionality and sample size are comparable is conducted. Using random matrix theory, the misclassification probability is shown to converge to a quantity that is a function of the true statistics of the data. Additionally, an estimator of the misclassification probability is derived. Finally, computationally efficient tuning of the parameter using this estimator is demonstrated on real data.

INDEX TERMS Estimation noise, LDA, linear classifier, MMSE, random matrix theory, weight vector.

I. INTRODUCTION

A binary linear classifier classifies a data point to one class or the other by thresholding a discriminant that is a linear combination of the data features. The weights of the features make up a *weight vector* and the constant term in the discriminant is the *bias* of the classifier.

Despite the availability of sophisticated non-linear methods for classification, linear classifiers are still widely used. In fact, new variants of standard linear methods catering to specific settings and applications are being developed all the time. A search of the recent literature reveals that linear classifiers are being employed in many tasks including clinical neuroimaging [2], digital pulse shape discrimination [3], predicting the genetic merit of beef cattle [4], and in conjunction with other methods for applications such as pathogen identification [5], strategy representation [6], and cancer classification [7]. Linear classifiers are especially suited to certain high-dimensional datasets on which they perform comparably with non-linear classifiers, with the advantage of much faster training times and quicker classification [8]. Due to ease of computation, linear classifiers further make good trial classifiers during the initial exploratory phase, when the relationship between the data features and labels is yet unknown [9].

One way of improving a given linear classifier's performance on a particular dataset is by tuning its bias so as to minimize training error on that dataset [10]. Because the bias is a scalar, a grid search for the optimum is computationally

undemanding. Even the need for a grid-search can be eliminated in many cases for which explicit representations of the optimal bias can be derived. For example, the authors of [11] derive an explicit bias correction of the Linear Discriminant Analysis (LDA) classifier discriminant in order to improve classification in the high estimation noise regime. The authors of [12] similarly correct for the bias of this classifier in an explicit form, but in the context of cost-sensitive classification. Additionally, the references [13] and [14] provide explicit bias corrections for certain high-dimensional variants of LDA. A related question has to do with improving upon a linear classifier's weight vector, which cannot be tuned or corrected in the same way. Relying on the intuition that a good weight vector should be able to extract the maximum discriminatory information content from the data point being classified, we show in this work that tuning the multidimensional weight vector can indeed be reduced to tuning a scalar.

In the first half of this paper, it is shown that any binary linear classifier discriminant can be decomposed into terms containing discriminating information and non-discriminating noise. A linear form of this decomposition parameterized by a variable α controls the trade-off between conflicting noise and information terms. At the optimal setting of α, the modified discriminant performs at least as good as the original classifier from which it was produced. Following this, the effect of the weight vector modification on the performance of an assortment of linear classifiers under different data dimensionality and sample size scenarios is studied. The method specifically yields significant performance gains for the Linear Discriminant Analysis (LDA) classifier under high estimation noise. Interestingly, the parameterized LDA operates as a bridge between LDA and the nearest centroid classifier, and performs at least as good as either of these classifiers. Additionally, it is shown that tuning the weight vector according to the proposed method can significantly improve the performance of certain classifiers whose native hyperparameters are not optimally set. It is shown that with weight vector tuning, the Support Vector Machine (SVM) with non-optimally tuned penalty can achieve performance close to that of its tuned counterpart. In this case, tuning the weight vector is fundamentally different from tuning the native hyperparameter of the classifier as it occurs post weight vector generation, while the native hyperparameter tuning occurs prior to weight vector generation. For SVM, generating the weight vector for each value of the native hyperparameter involves solving an optimization problem. Tuning the weight vector according to the proposed method, however, reduces to a simple grid search over a scalar parameter. This idea can be generalized to any classifier with hyperparameters that are set prior to weight vector generation.

The second half of the paper consists of an asymptotic study of the parameterized LDA classifier under a growth regime in which the data dimensionality and sample size grow proportionally. Under a Gaussian assumption on the data distribution, we use random matrix theory to show that the probability of misclassification of this classifier converges to a limit that is a function of the true class statistics. We also derive a consistent estimator of the probability of misclassification by which the classifier parameter α can be tuned. This estimator is more computationally efficient than other tuning methods which rely on additional testing points or recycling the training set, e.g. cross-validation, as it requires no additional testing points and no averaging. We demonstrate its performance on real data.

An additional finding of this work is a new interpretation of the optimality of LDA. The LDA decision rule, derived by maximizing the posterior probability of a test point, assuming that it is drawn from a Gaussian distribution with classes having distinct means and common covariances, yields a weight vector which is the optimal Bayes direction. It can be shown that, under a common class covariance, the weight vector resulting from Fisher's linear discriminant, in which the ratio of the distance between the projected class means and the within class variance is maximized, is proportional to the Bayes direction [10]. A proportional solution can also be arrived at via a least squares formulation of the fitted data from their labels in the binary case [10]. This makes the Bayes direction optimal in the posterior probability sense, the Fisher's linear discriminant sense, and the least squares sense. Moreover, this paper shows that the Bayes direction is optimal in the sense that it achieves the minimum noise (in the mean square error sense) with respect to the test point when the classes are Gaussian with common covariance.

To summarize, the main contributions of this paper are

- A practical method for weight vector tuning which reduces to grid search over a scalar parameter.
- A novel interpretation of the optimality of the LDA classifier in terms of minimizing test point noise.
- Asymptotic expressions for the probability of misclassification of the parameterized LDA classifier.
- A consistent estimator of the probability of misclassification of the parameterized LDA classifier.

II. WEIGHT VECTOR TUNING PROCEDURE

Consider a supervised classification problem in which a test point $\mathbf{x} \in \mathbb{R}^p$ is to be labeled as belonging to one of two classes $\mathcal{C}_0$ and $\mathcal{C}_1$. A linear classification approach to this problem imposes a discriminant of the form

$$\mathbf{w}^T\mathbf{x} + w_0, \tag{1}$$

characterized by a weight vector, $\mathbf{w} \in \mathbb{R}^p$, and bias, $w_0 \in \mathbb{R}$. The decision rule $C(\mathbf{x}) = \mathbb{1}\{\mathbf{w}^T\mathbf{x} + w_0 > 0\}$ based on (1) then classifies $\mathbf{x}$ to one of the two classes, i.e., $C(\mathbf{x}) = i$ indicates that $\mathbf{x}$ is classified to class $\mathcal{C}_i$, $i = 0, 1$. Examples of classifiers which fit this form include LDA, SVM and Least-Squares SVM (both using linear kernels), and Regularized LDA (R-LDA).

In this paper, we propose a method of tuning the weight vector $\mathbf{w}$, which reduces the non-discriminative 'noisy' components of the original discriminant (1). As a result, the modified discriminant achieves a testing error rate at least as good as the original and, in certain cases, much better.

Throughout this paper, let the means and covariances of classes $\mathcal{C}_0$ and $\mathcal{C}_1$ be denoted by $\boldsymbol{\mu}_0$, $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_1$ respectively. In Section II-A, we explore an ideal case in which the discriminant neatly decomposes into separate information and noise terms and the noises cancel out optimally in a linear fashion under the assumption of perfectly known means and that $\mathcal{C}_0$ and $\mathcal{C}_1$ makeup a Gaussian mixture model with common class covariance $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$. Inspired by the findings of Section II-A, in Section II-B we heuristically extend this result to a more practical scenario which assumes unknown means and no restriction on the class distributions.

A. KNOWN CLASS MEANS

In this section, assume that the data distribution means $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ are known exactly and that $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$. We proceed to derive a noise-minimized version of (1).

Consider the shifted test point $\tilde{\mathbf{x}} = \mathbf{x} - \frac{\boldsymbol{\mu}_0+\boldsymbol{\mu}_1}{2}$. For any given classifier with weight vector $\mathbf{w}$, we show that the projection of $\tilde{\mathbf{x}}$ onto $\mathbf{w}$, i.e., $\mathbf{w}^T\tilde{\mathbf{x}}$, can be decomposed into 'informative' components which aid in discriminating the class of $\mathbf{x}$ and 'noisy' components which interfere with discriminating the class of $\mathbf{x}$. We then take advantage of this hidden structure for the purpose of reducing the overall noise and obtaining a better classifier.

Let $\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. The expression $\tilde{\mathbf{x}}$ can be expressed as the sum of its projection onto $\boldsymbol{\mu}$ and projection orthogonal to $\boldsymbol{\mu}$ as

$$\tilde{\mathbf{x}} = \frac{\boldsymbol{\mu}\boldsymbol{\mu}^T}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\tilde{\mathbf{x}} + \mathbf{P}_{\boldsymbol{\mu}}\tilde{\mathbf{x}} \tag{2}$$

where $\mathbf{P}_{\boldsymbol{\mu}} = \left(\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^T}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\right)$ is the projection orthogonal to $\boldsymbol{\mu}$. Substituting (2) into $\mathbf{w}^T\tilde{\mathbf{x}}$ results in the decomposition of $\mathbf{w}^T\tilde{\mathbf{x}}$ as

$$\frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu}^T\tilde{\mathbf{x}} + \mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\tilde{\mathbf{x}} \tag{3}$$

We now show that the first term in (3) is composed of an informative component and noisy component with respect to $\mathbf{x}$, while the second term consists solely of noise. Assume $\mathbf{x} \in \mathcal{C}_i$, where i is either 0 or 1. Then, assuming the Gaussian mixture model

$$\mathbf{x}|\mathbf{x} \in \mathcal{C}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}) \tag{4}$$

with i^{th} class prior $\pi_i = P[\mathbf{x} \in \mathcal{C}_i]$, we have $\mathbf{x}|\mathbf{x} \in \mathcal{C}_i \sim \boldsymbol{\mu}_i + \boldsymbol{\Sigma}^{1/2}\mathbf{z}$, $i = 0, 1$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The first term in (3), conditioned on the class of $\mathbf{x}$, is then distributed as follows

$$\begin{aligned}\frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu}^T\tilde{\mathbf{x}}|\mathbf{x} \in \mathcal{C}_i &\sim \frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu}^T\left((-1)^{i+1}\frac{\boldsymbol{\mu}}{2} + \boldsymbol{\Sigma}^{1/2}\mathbf{z}\right) \\ &= \underbrace{(-1)^{i+1}\frac{\mathbf{w}^T\boldsymbol{\mu}}{2}}_{I_1(\text{information})} + \overbrace{\frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{1/2}\mathbf{z}}^{N_1(\text{noise})}\end{aligned} \tag{5}$$

The first term in (5) carries information about the class of $\mathbf{x}$ through its sign. The second term is the same regardless of the class of $\mathbf{x}$ and therefore carries no discriminating information. This is a direct result of assuming a common covariance between $\mathcal{C}_0$ and $\mathcal{C}_1$. The informative component is denoted by I_1 while the noisy component with respect to $\mathbf{x}$ is denoted by N_1. Similarly,

$$\begin{aligned}\mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\tilde{\mathbf{x}}|\mathbf{x} \in \mathcal{C}_i &\sim \mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\left((-1)^{i+1}\frac{\boldsymbol{\mu}}{2} + \boldsymbol{\Sigma}^{1/2}\mathbf{z}\right) \\ &= \underbrace{\mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\boldsymbol{\Sigma}^{1/2}\mathbf{z}}_{N_2(\text{noise})}\end{aligned}$$

The discriminatory component of this term is lost in the orthogonal projection, and therefore this term consists solely of noise with respect to the testing point, denoted by N_2.

To recap, the decomposition of the weight vector divides the discriminant into a single observable term containing I_1 and N_1 and a single observable term containing N_2. Without the decomposition, none of these individual noise/information terms are accessible. Now, in the interest of achieving better classification performance, we wish to reduce the overall noise content in the discriminant. We can leverage the observable term containing N_2 to bring out the information in the observable term containing both information I_1 and noise N_1. To this end, consider the following modification of the discriminant (3),

$$\frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu}^T\tilde{\mathbf{x}} + g\left(\mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\tilde{\mathbf{x}}\right) \tag{6}$$

for any function $g(\cdot)$, and which, by the above analysis, is equivalent to

$$I_1 + N_1 + g(N_2)$$

The optimal $g(\cdot)$ such that

$$\mathbb{E}\left[(N_1 + g(N_2))^2\right]$$

is minimized is the MMSE estimator $\mathbb{E}[-N_1|N_2]$. This choice of $g(\cdot)$ has the effect of minimizing the total noise in the discriminant in the mean square error sense. We show in Section II-A1 that it simultaneously minimizes the probability of misclassification. In the following Lemma 1, we derive the exact form of $g(\cdot)$ for a given $\mathbf{w}$ based on the class distribution assumptions (4).

Lemma 1: The optimal $g(N_2)$ is the linear function of N_2 given by $g^*(N_2) = \alpha_{\text{MMSE}}(\mathbf{w})N_2$, where

$$\alpha_{\text{MMSE}}(\mathbf{w}) = -\frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\frac{\boldsymbol{\mu}^T\boldsymbol{\Sigma}\mathbf{P}_{\boldsymbol{\mu}}\mathbf{w}}{\mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\boldsymbol{\Sigma}\mathbf{P}_{\boldsymbol{\mu}}\mathbf{w}}. \tag{7}$$

Proof: See the proof of Lemma 1 in our extended report available on arXiv [1]. Note that N_2 is observable only through the expression $\mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\tilde{\mathbf{x}}$ and so when using this result we replace N_2 by its observable counterpart. Based on this result, we have the following theorem.

Theorem 1: The discriminant that minimizes the noise with respect to the test point in the MSE sense for a given $\mathbf{w}$, known

means, and under the data distribution assumptions of (4), is

$$\frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu}^T\tilde{\mathbf{x}} + \alpha_{\text{MMSE}}(\mathbf{w})\mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\tilde{\mathbf{x}}, \tag{8}$$

or, equivalently, $\mathbf{w}'^T\mathbf{x} + w_0'$, where

$$\mathbf{w}' = \frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu} + \alpha_{\text{MMSE}}(\mathbf{w})\mathbf{P}_{\boldsymbol{\mu}}\mathbf{w}$$

and

$$w_0' = -\frac{1}{2}\left(\frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu} + \alpha_{\text{MMSE}}(\mathbf{w})\mathbf{P}_{\boldsymbol{\mu}}\mathbf{w}\right)^T(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1).$$

Theorem 1 is obtained by simply evaluating (6) using $g^*(\cdot)$. Note that this result applies to a general weight vector $\mathbf{w}$ in a known mean setting. Section II-B1 considers actual weight vectors corresponding to popular linear classifiers in an unknown mean setting, while Section III studies the weight vector tuning of LDA, in particular, in an unknown mean setting.

We now make several remarks concerning Theorem 1. Firstly, the modified discriminant is linear. This is a direct result of the Gaussian assumption (4), which, while not technically necessary, is desirable, as it produces a simple linear form which inspires the parameterized formulation presented in the next section. Secondly, the original weight vector $\mathbf{w}$ is modified to $\mathbf{w}'$ and a bias w_0' is generated. This bias is the optimal bias in the sense of minimizing the probability of misclassification under the class distribution assumptions of (4) and equal class priors when fixing the weight vector to $\mathbf{w}'$ (see [15] Proposition 2). Finally, viewing the modified discriminant (8) as a function of a parameter α as follows

$$\frac{\mathbf{w}^T\boldsymbol{\mu}}{\boldsymbol{\mu}^T\boldsymbol{\mu}}\boldsymbol{\mu}^T\tilde{\mathbf{x}} + \alpha\mathbf{w}^T\mathbf{P}_{\boldsymbol{\mu}}\tilde{\mathbf{x}}, \tag{9}$$

$\alpha = \alpha_{\text{MMSE}}(\mathbf{w})$ yields a stationary point of its probability of misclassification and achieves the minimum probability of misclassification when $\mathbf{w}^T\boldsymbol{\mu} > 0$. This is demonstrated in Section II-A1.

The following corollary of Theorem 1 lends intuition as well as credibility to this technique by showing that it recovers the Bayes optimal classifier discriminant for the assumed class distributions from its weight vector. The Bayes classifier in this case is linear. It is the LDA classifier, with decision rule

$$\mathbb{1}\left\{\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{x}} + \ln\frac{\pi_1}{\pi_0} > 0\right\}. \tag{10}$$

The LDA weight vector is $\mathbf{w} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$.

Corollary 1: Computing the parameter (7) corresponding to the LDA classifier (10) yields

$$\alpha_{\text{MMSE}}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right) = 1$$

and the resulting discriminant (8) recovers the LDA discriminant in (10) when the class priors are equal.

Since there is no modification of the weight vector, we conclude that the LDA weight vector (in the case of known statistics) is optimal relative to itself in that it achieves the minimum noise (in the mean square error sense) with respect to the test point under the assumed class distributions.

1) EXPERIMENTS WITH KNOWN MEANS

For the following simulation and any simulations involving synthetic data in the remainder of this paper, the exact expected testing error/probability of misclassification of a linear classifier learned on a given training set is computed using knowledge of the data distribution from which the testing data is generated. All synthetic data in this paper is generated from a two-class Gaussian mixture model. The expected testing error under these data distribution assumptions of a generic binary linear classifier

$$\mathbb{1}\left\{\boldsymbol{\beta}^T\mathbf{x} + \beta_0 > 0\right\},$$

with weight vector $\boldsymbol{\beta}$ and intercept β_0, can easily be derived as (see Lemma 1 in [16])

$$\pi_0\Phi\left(\frac{\boldsymbol{\beta}^T\boldsymbol{\mu}_0 + \beta_0}{\sqrt{\boldsymbol{\beta}^T\boldsymbol{\Sigma}_0\boldsymbol{\beta}}}\right) + \pi_1\Phi\left(-\frac{\boldsymbol{\beta}^T\boldsymbol{\mu}_1 + \beta_0}{\sqrt{\boldsymbol{\beta}^T\boldsymbol{\Sigma}_1\boldsymbol{\beta}}}\right). \tag{11}$$

Now consider the parameterized version (9) of (8). The objective of the following simulation is to show that $\alpha_{\text{MMSE}}(\mathbf{w})$ given by (7) coincides with the α yielding a stationary point of the expected testing error of (9). The stationary point is a minimum when $\mathbf{w}^T\boldsymbol{\mu} > 0$ and is otherwise a maximum, as in that case, the orientation of $\mathbf{w}$ flips the class labels.

To demonstrate this, a weight vector $\mathbf{w}$ is uniformly sampled from all $\mathbf{w}$ such that $\|\mathbf{w}\|_2 = 1$ using the method in [17]. It is then fed to (9) and the exact expected testing error with varying α is plotted using (11). The quantity $\alpha_{\text{MMSE}}(\mathbf{w})$ is then computed from (7) for comparison. The class statistics used for this simulation are

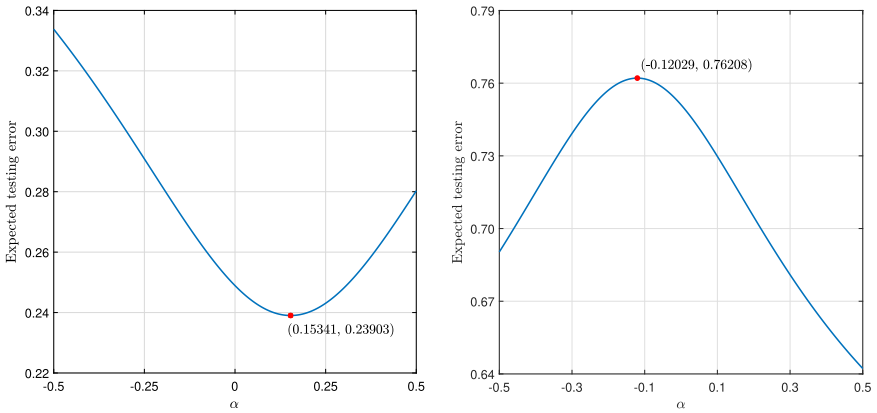
$$\boldsymbol{\mu}_0 = \frac{1}{p^{1/4}}\left[\mathbf{1}^T_{\lceil\sqrt{p}\rceil}\ \mathbf{0}^T_{p-\lceil\sqrt{p}\rceil-2}\ 2\ 2\right]^T,\ \boldsymbol{\mu}_1 = \mathbf{0}_p, \tag{12}$$

and

$$\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \frac{10}{p}\mathbf{1}_p\mathbf{1}_p^T + 0.1\mathbf{I}_p \tag{13}$$

where $p = 200$. Here, $\pi_0 = \pi_1 = 0.5$.

Fig. 1(a) and (b) show the results when $\mathbf{w}^T\boldsymbol{\mu} > 0$ and $\mathbf{w}^T\boldsymbol{\mu} < 0$, respectively. In Fig. 1(a), the minimum expected testing error occurs at $\alpha = 0.15341$. This exactly coincides with $\alpha_{\text{MMSE}}(\mathbf{w})$ of Theorem 1 that minimizes the noise in the discriminant. In Fig. 1(b), the *maximum* expected testing error occurs at $\alpha = -0.12029$, which, again, exactly coincides with $\alpha_{\text{MMSE}}(\mathbf{w})$ that minimizes the noise in the discriminant. The latter discriminant's behavior can be explained by the fact that the orientation of the randomly generated $\mathbf{w}$ flips the class labels. Simply taking the negative of $\mathbf{w}$ yields a classifier having the *minimum* expected testing error at $\alpha_{\text{MMSE}}(\mathbf{w})$. In conclusion, minimizing the noise in the discriminant in the MSE sense is equivalent to minimizing the expected testing error, as long as $\mathbf{w}$ is sensibly oriented. This motivates using



(a) Here $\mathbf{w}^T\boldsymbol{\mu} > 0$ and $\alpha_{\text{MMSE}}(\mathbf{w}) = 0.1534$ coincides with the α yielding the minimum expected testing error

(b) Here $\mathbf{w}^T\boldsymbol{\mu} < 0$ and $\alpha_{\text{MMSE}}(\mathbf{w}) = -0.1203$ coincides with the α yielding the maximum expected testing error

FIGURE. 1. Plot of the expected testing error of (9) against α for a randomly generated weight vector w.

this criteria as the basis for designing a better classifier in the next section.

B. UNKNOWN CLASS MEANS

The previous section derives the discriminant with minimum noise with respect to the test point for a general binary linear classifier with weight vector $\mathbf{w}$ under the assumption of Gaussian classes with known means and a common covariance. A more practical scenario is when all class statistics are unknown and sample statistics are used instead. Using the sample mean estimates introduces an additional estimation noise into the discriminant.

Let the n individual training vectors corresponding to classes $\mathcal{C}_0$ and $\mathcal{C}_1$ make up the columns of the matrices $\mathbf{X}_0 \in \mathbb{R}^{p\times n_0}$ and $\mathbf{X}_1 \in \mathbb{R}^{p\times n_1}$, respectively ($n = n_0 + n_1$). The maximum likelihood estimates of the class means are given by the sample means $\hat{\boldsymbol{\mu}}_0 = \frac{1}{n_0}\mathbf{X}_0\mathbf{1}_{n_0}$ and $\hat{\boldsymbol{\mu}}_1 = \frac{1}{n_1}\mathbf{X}_1\mathbf{1}_{n_1}$. Let $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_0$ and $\hat{\tilde{\mathbf{x}}} = \mathbf{x} - \frac{\hat{\boldsymbol{\mu}}_0+\hat{\boldsymbol{\mu}}_1}{2}$. Given a weight vector, $\mathbf{w}$, $\mathbf{w}^T\hat{\tilde{\mathbf{x}}}$ can be expressed as

$$\mathbf{w}^T\hat{\tilde{\mathbf{x}}} = \frac{\mathbf{w}^T\hat{\boldsymbol{\mu}}}{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}\hat{\boldsymbol{\mu}}^T\hat{\tilde{\mathbf{x}}} + \mathbf{w}^T\mathbf{P}_{\hat{\boldsymbol{\mu}}}\hat{\tilde{\mathbf{x}}} \tag{14}$$

where $\mathbf{P}_{\hat{\boldsymbol{\mu}}} = (\mathbf{I} - \frac{\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^T}{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}})$. Regardless of the class distributions and whether assuming distinct covariances $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ or common class covariances $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$, following a similar line of logic to the analysis in Section II-A reveals that, while the first term in (14) is similarly composed of both information and noise (whether that be estimation noise, noise from the test point, or both), the second term is not purely noise. In fact, it is informative. This is shown in detail in Appendix A of our extended report available on arXiv [1].

Thus, when the means are unknown, the approach taken in Section II-A of minimizing the squared sum of 'noise 1' with the second term no longer applies, as the second term is informative. Nonetheless, the interaction of this term with the noise in the first term can potentially yield performance gains and so motivated by Section II-A, the following parameterized version of the sample statistic equivalent of (8) is proposed

$$\frac{\mathbf{w}^T\hat{\boldsymbol{\mu}}}{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}\hat{\boldsymbol{\mu}}^T\hat{\tilde{\mathbf{x}}} + \alpha\mathbf{w}^T\mathbf{P}_{\hat{\boldsymbol{\mu}}}\hat{\tilde{\mathbf{x}}} \tag{15}$$

where α is a parameter to be tuned.

The following Section II-B1 demonstrates that a better misclassification rate may be achieved by setting α to a value that is not equal to one (where $\alpha = 1$ recovers the original projection with optimal bias assuming equal priors and the class distribution in (4)). A significant improvement is observed when the estimation noise is high.

1) EXPERIMENTS WITH UNKNOWN MEANS

In this section we explore the behavior of (15) under a variety of settings and for an assortment of starting weight vectors. We first list and briefly describe the discriminants from which these weight vectors are extracted, namely, LDA, logistic regression, linear support vector machine (SVM), regularized LDA (R-LDA), and randomly-projected LDA ensemble (RP-LDA).

- **LDA** (see [10]) in the form (10) is the Bayes classifier for data distributed as (4). In practice, the class statistics are unknown and sample estimates are used instead. The sample means $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\mu}}_1$ are defined at the beginning of Section II-B. The maximum likelihood estimates of the common covariance matrix and class priors are the pooled sample covariance matrix

$$\hat{\boldsymbol{\Sigma}} = \frac{(n_0-1)\hat{\boldsymbol{\Sigma}}_0 + (n_1-1)\hat{\boldsymbol{\Sigma}}_1}{n_0+n_1-2},$$

where $\hat{\boldsymbol{\Sigma}}_0 = \frac{1}{n_0-1}(\mathbf{X}_0 - \hat{\boldsymbol{\mu}}_0\mathbf{1}^T)(\mathbf{X}_0 - \hat{\boldsymbol{\mu}}_0\mathbf{1}^T)^T$ and $\hat{\boldsymbol{\Sigma}}_1 = \frac{1}{n_1-1}(\mathbf{X}_1 - \hat{\boldsymbol{\mu}}_1\mathbf{1}^T)(\mathbf{X}_1 - \hat{\boldsymbol{\mu}}_1\mathbf{1}^T)^T$, and the prior estimates $\hat{\pi}_i = \frac{n_i}{n}$, $i = 0, 1$, respectively. The LDA discriminant is

then

$$\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\tilde{\boldsymbol{x}}} + \ln \frac{\hat{\pi}_1}{\hat{\pi}_0}.$$

- For linearly separable training data, **SVM with linear kernel** (see [10]) finds a hyperplane that maximizes the margin between one class and the other subject to constraints of perfect classification on the training points. When the training data is linearly inseparable, the constraints are relaxed by penalizing each (possibly) misclassified point. The penalty is a parameter that must be tuned. This variant is called the soft-margin SVM with linear kernel, and it is what we use in this paper.
- **Logistic regression** (see [10]) models the log-odds '$\ln(\frac{P[\mathbf{x}\in\mathcal{C}_1|\mathbf{x}]}{1-P[\mathbf{x}\in\mathcal{C}_1|\mathbf{x}]})$' as a linear function of the test point. The decision boundary corresponds to the set of points at which the log-odds equals zero. The weight vector and bias of the decision boundary are learned by maximizing the likelihood of the training data.
- **R-LDA** counters the small sample issue in LDA by regularizing the pooled sample covariance estimate before inverting it. There are several possibilities for the form of the regularization (see [18]). In this paper we opt for

  $$\hat{\boldsymbol{\mu}}^T \left(\hat{\boldsymbol{\Sigma}} + \gamma \mathbf{I}_p\right)^{-1} \hat{\tilde{\boldsymbol{x}}} + \ln \frac{\hat{\pi}_1}{\hat{\pi}_0},$$

  where γ is the regularization parameter that must be tuned.
- **RP-LDA ensemble** (see [19]) counters the small sample issue in LDA by reducing the dimensionality of the training samples (and test point) using random matrices. Each projection $\mathbf{R}_i \in \mathbb{R}^{d\times p}$ yields a discriminant. These are averaged over all M projections so that the final discriminant has the form

  $$\frac{1}{M}\sum_{i=1}^{M} \hat{\boldsymbol{\mu}}^T \mathbf{R}_i^T (\mathbf{R}_i \hat{\boldsymbol{\Sigma}} \mathbf{R}_i^T)^{-1} \mathbf{R}_i \hat{\tilde{\boldsymbol{x}}} + \ln \frac{\hat{\pi}_1}{\hat{\pi}_0}$$

  The reduced dimension d is a parameter that must be tuned.

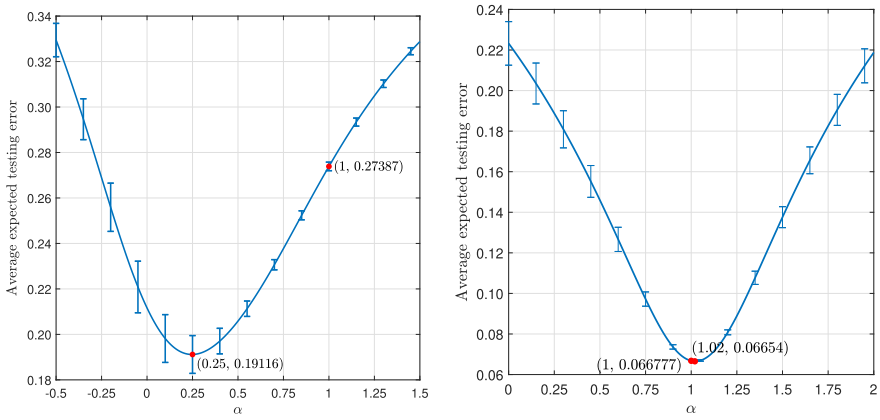
For these simulations, we consider two data distributions: data generated from classes having a common covariance and data generated from classes having distinct covariance matrices. We also consider three regimes of n versus p: n on the order of p ($p = 400$, $n = 450$), $n > p$ ($p = 10$, $n = 500$), and $n < p$ ($p = 300$, $n = 100$). We apply the appropriate classifiers to each regime. LDA requires $n > p$, soft-margin SVM is applicable in any regime, logistic regression requires n be much greater than p to ensure convergence of the maximum likelihood estimates of the weight vector and bias, and finally, R-LDA and RP-LDA are designed for the regime $n < p$.

Each classifier is trained on a generated training set. Additionally, for SVM, R-LDA, and RP-LDA, the penalty, γ, and d parameters are chosen to minimize the expected testing error given that training set. The SVM penalty is tuned within the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000\}$, γ within the set $[10^{-4}, 2]$, in increments of 0.1, and d from 1 to the maximum allowable setting of $d = \text{rank}(\hat{\boldsymbol{\Sigma}}) - 2$, in increments of 2. After this is done, we have a weight vector $\mathbf{w}$ for each classifier. Each weight vector is fed into (15) to obtain an α-parameterized version of the discriminant. Let us refer to these new classifiers as α-LDA, α-SVM, α-log, α-RLDA, and α-RPLDA for short. For each α-parameterized discriminant, we vary α and compute the expected testing error using (11). These errors are averaged over 100 independently generated training sets. Error bars depicting the standard errors are plotted alongside this average.

Recall that setting $\alpha = 1$ in (15) produces a discriminant having the original weight vector $\mathbf{w}$ and a bias with minimum probability of misclassification (under the Gaussian mixture model and equal priors assumption) for that weight vector. In what follows, we use $\alpha = 1$ as a reference point for determining whether or not there is a significant improvement in classifier performance at the α achieving the minimum error rate. To quantify the improvement, we report percentage changes relative to the average expected testing error at $\alpha = 1$ computed as $\frac{\text{error at } \alpha \text{ achieving the minimum} - \text{error at } \alpha=1}{\text{error at } \alpha=1} \times 100$. This quantity reflects the fact that a given error improvement starting at an already low error rate at the baseline $\alpha = 1$ is more significant than when the error is high to start with.

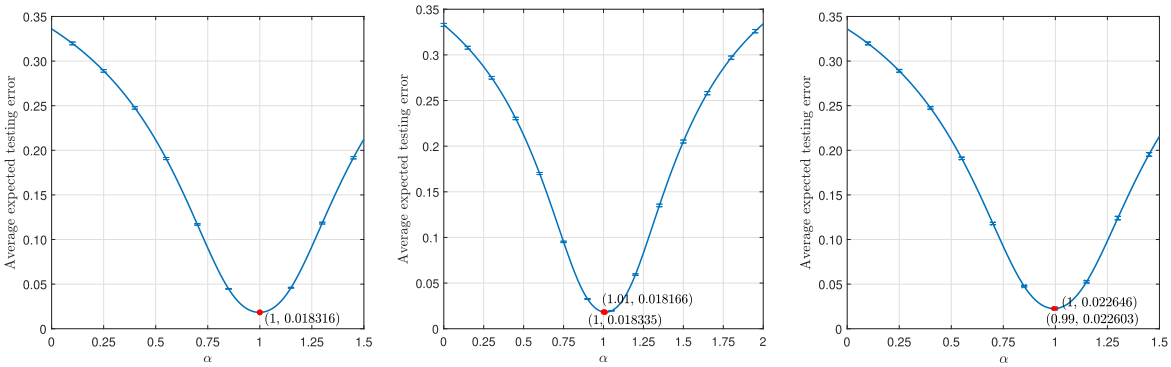
The first set of class statistics we consider are (12), (13), and $\pi_0 = \pi_1 = 0.5$. Corresponding to this data distribution are Figs. 2, 3, and 4.

Fig. 2(a) and (b) plot the average expected testing errors of α-LDA and α-SVM respectively against varying α when $p = 400$ and $n = 450$. At $\alpha = 0.25$, the α-LDA classifier achieves a 30.2% relative decrease in the average expected testing error. Note that ordinary LDA ($\alpha = 1$) is nowhere near optimal. On the other hand, α-SVM achieves a 0.355% decrease in average expected testing error at $\alpha = 1.02$. These results suggest that there is a lot to be gained performance-wise by LDA in this regime but not so much by linear SVM. This can be attributed to the fact that LDA relies on sample estimation and that the noise due to estimation is high when $p = 400$ and $n = 450$. This is further supported by the results of Fig. 3(a), (b) and (c), which plot the average expected testing errors of α-LDA, α-SVM, and α-log, respectively against varying α when $p = 10$ and $n = 500$. The minimum average expected occurs at exactly $\alpha = 1$ for α-LDA, $\alpha = 1.01$ for α-SVM and at $\alpha = 0.99$ for α-log, with the latter two classifiers achieving a relative decrease of no more than 1% and 0.2% respectively. The extreme behavior in all three figures can be explained by the fact that there is very little estimation noise for this choice of dimensions. What is notable is the difference between Figs. 2(a) and 3(a) whch suggests that the weight vector tuning method is most effective under high estimation noise and for methods which are most sensitive to it. This idea is again reinforced in Fig. 4(a), (b), and (c), in which the average expected testing errors of α-RLDA, α-RPLDA, and α-SVM respectively are plotted against varying α when $p = 300$ and $n = 100$. The relative decrease in errors for each of the three



(a) α-LDA (b) α-SVM

FIGURE. 2. Plots of expected testing error averaged over 100 training sets for data generated from classes with a common Σ. Here, $p = 400$ and $n = 450$.



(a) α-LDA (b) α-SVM (c) α-log

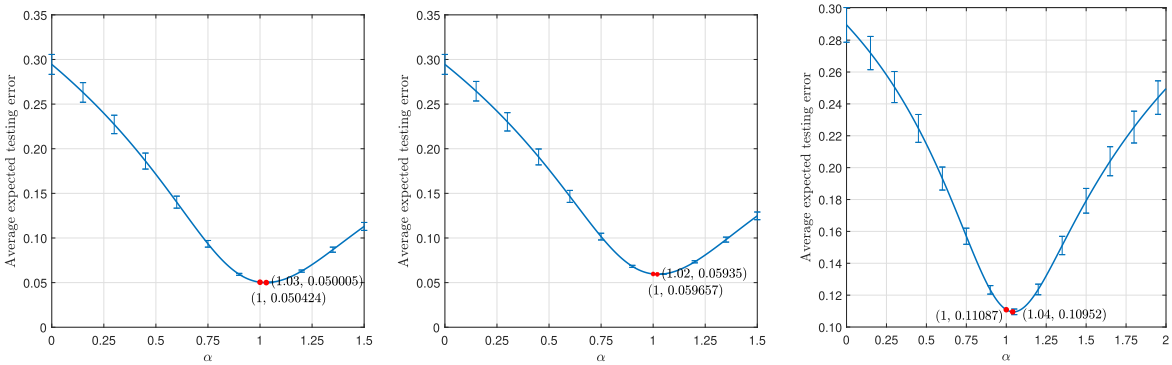
FIGURE. 3. Plots of expected testing error averaged over 100 training sets for data generated from classes with a common Σ. Here, $p = 10$ and $n = 500$.

classifiers does not exceed 1.3%. It must be that R-LDA and RP-LDA are able to reduce much of the estimation noise on their own, and so the α parameterization does not bring much improvement.

Figs. 5 and 6 are based on data with the class statistics (12), $[\mathbf{\Sigma}_0]_{ij} = 0.9^{|i-j|}$, $i, j = 1, \ldots, p$, $\mathbf{\Sigma}_1 = \frac{10}{p}\mathbf{1}_p\mathbf{1}_p^T + 0.1\mathbf{I}_p$, and $\pi_0 = \pi_1 = 0.5$. The difference here is that the class covariances are distinct. Fig. 5(a) and (b) again plot the average expected testing errors of α-LDA and α-SVM, respectively, against varying α when $p = 400$ and $n = 450$. In this case, α-LDA significantly improves in performance when α is set to a non-unit value. It achieves a relative decrease in error of 27.6% at $\alpha = 0.05$, while $\alpha-$SVM achieves a relative decrease in error of 0.4% at $\alpha = 1.14$. Finally, Fig. 6(a), (b), and (c) plot the average expected testing errors of α-RLDA, α-RPLDA and α-SVM against varying α when $p = 300$ and $n = 100$. Here, the relative decreases in error do not exceed 0.6%.

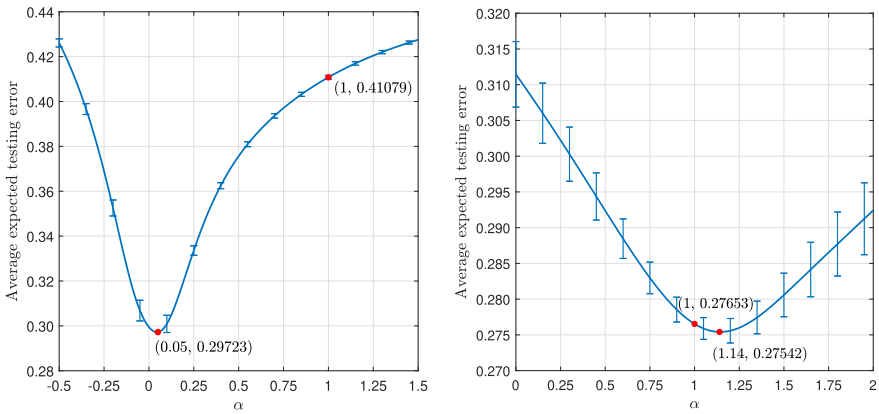
As described at the beginning of this section, for each training set, the SVM penalty is tuned to the value yielding the lowest expected testing error. We found that SVM does not show much improvement when it is α parameterized. It is interesting to observe what happens when the penalty is not tuned beforehand. Instead we set the penalty to 1 (its default setting in the MATLAB R2019b 'fitcsvm' function) uniformly across all training sets. Fig. 7 shows the resulting average expected testing error of α-SVM plotted against vary α in the same setting as in Fig. 5(b), i.e. $p = 450$, $n = 400$, and distinct $\mathbf{\Sigma}_0$ and $\mathbf{\Sigma}_1$. In this case, α-SVM achieves a relative decrease in error of 17.7% at $\alpha = 0.2$. Clearly, the method improves performance when $\mathbf{w}$ itself is not at its optimal.

Taking this idea further, we show that tuning the weight vector of a SVM classifier with a poorly chosen penalty can compensate for the resulting loss in performance. Fig. 8 is based on the USPS dataset consisting of separate training and testing sets of grayscale images of handwritten digits $0 - 9$. Pairs of digits are used to form a binary classification problem. For each pair of digits, a poorly tuned SVM classifier is α parameterized and the testing error plotted against α to illustrate the effect of weight vector tuning.



(a) α-RLDA (b) α-RPLDA (c) α-SVM

FIGURE. 4. Plots of expected testing error averaged over 100 training sets for data generated from classes with a common Σ. Here, $p = 300$ and $n = 100$.



(a) α-LDA (b) α-SVM

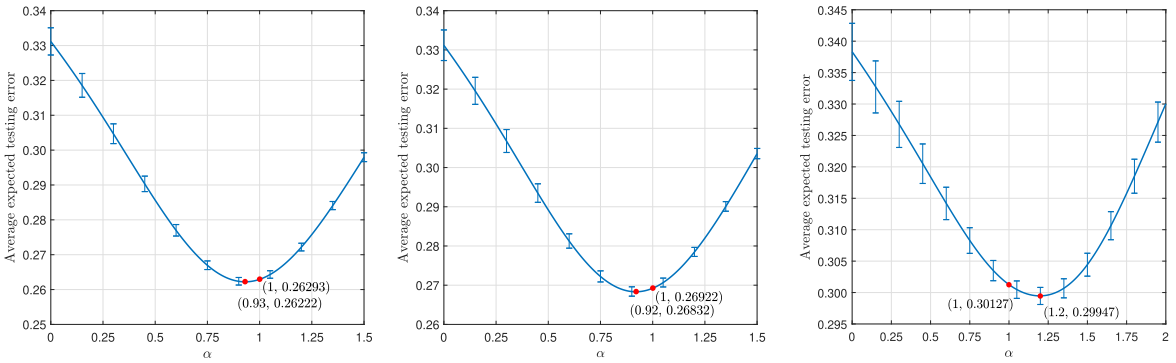
FIGURE. 5. Plots of expected testing error averaged over 100 training sets for data generated from classes with distinct Σ_0 and Σ_1. Here, $p = 400$ and $n = 450$.

For the digit pair '2' and '6,' an optimized SVM classifier can achieve a testing error of 0.0217. Fig. 8(a) shows the testing error of α-SVM starting with a poorly tuned SVM classifer whose testing error on this digit pair is 0.0489. By weight vector tuning, the testing error can be brought down to 0.0272. This is comparable to the performance of the original optimized SVM classifer. Similarly, for the digit pair '3' and '5,' an optimized SVM classifier can achieve a testing error of 0.0675. Fig. 8(a) shows the testing error of α-SVM starting with a poorly tuned SVM classifer whose testing error on this digit pair is 0.0951. By weight vector tuning, the testing error can be brought down to 0.0736.

The significance of this finding is the potential savings in computation that can be made by weight vector tuning versus penalty tuning. The reason for this is that weight vector tuning is an afterthought; it occurs post weight vector generation. On the other hand, setting the penalty is done prior to weight vector generation. An optimization problem must be solved to generate the weight vector with each setting of the penalty. At best, generating this weight vector has a complexity of $\mathcal{O}(n^2)$ at each setting of the penalty [20].

This idea generalizes to any linear classifier whose native hyperparameters are set prior to weight vector generation. The tuning of the hyperparameters will then involve repeatedly generating the weight vector. If this process is costly, weight vector tuning can provide a more computationally efficient method of improving performance than tuning the native hyperparameters. Another example that is not demonstrated here is the RP-LDA ensemble classifier whose projection dimension d is a native hyperparameter. Tuning this is computationally inefficient as it means projecting all the data with each setting of d. A simple alternative is weight vector tuning.

Before ending this section, we briefly touch on the question of multiple classes. The weight vector tuned classifier proposed in this work is a binary classifier. Nevertheless, it can be easily extended to a multi-class setting as in [14] which considers a variant of the LDA binary classifier derived under



(a) α-RLDA (b) α-RPLDA (c) α-SVM

FIGURE. 6. Plots of expected testing error averaged over 100 training sets for data generated from classes with distinct Σ_0 and Σ_1. Here, $p = 300$ and $n = 100$.

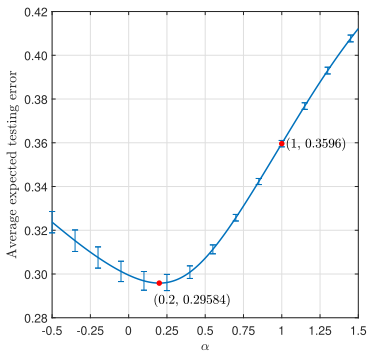


FIGURE. 7. Plot of expected testing error of α-SVM with penalty set to 1 averaged over 100 training sets for data generated from classes with distinct Σ_0 and Σ_1. Here, $p = 400$ and $n = 450$.

our same assumption of two Gaussian classes. To summarize, one may consider a one-versus-the-rest approach or a one-versus-one approach to apply a binary classifier to the multi-class setting [21]. Assuming each class is Gaussian, by grouping multiple classes together, the one-versus-the-rest approach violates this assumption as 'the rest' is a Gaussian mixture model. That leaves us with only the one-versus-one approach as a viable option, but this approach can lead to ambiguous classification [21]. As suggested in [14], these ambiguities can be resolved by favoring the class with the higher discriminant scores in the case of ties.

Overall, we conclude from this section that α-LDA in the 'n on the order of p' scenario shows the most promise in terms of improved performance. For this reason, we proceed to study this classifier in the RMT asymptotic regime in the next section.

III. ASYMPTOTIC ANALYSIS OF THE PARAMETERIZED LDA CLASSIFIER

In this section, we extend our study of α-LDA, the modified weight discriminant (8) corresponding to the plugin LDA weight vector. The α-LDA discriminant

$$\frac{\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}}{\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}}} \hat{\boldsymbol{\mu}}^T \hat{\tilde{\mathbf{x}}} + \alpha \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{P}_{\hat{\boldsymbol{\mu}}} \hat{\tilde{\mathbf{x}}}$$

is a bridging between LDA (when $\alpha = 1$) and the nearest centroid classifier (when $\alpha = 0$) with decision rule

$$\mathbb{1}\left\{||\hat{\boldsymbol{\mu}}_0 - \mathbf{x}||_2^2 - ||\hat{\boldsymbol{\mu}}_1 - \mathbf{x}||_2^2 > 0\right\} = \mathbb{1}\left\{\hat{\boldsymbol{\mu}}^T \hat{\tilde{\mathbf{x}}} > 0\right\}$$

which classifies $\mathbf{x}$ to the class with nearest sample mean. It is the Bayes classifier for data distributed as (4) when $\boldsymbol{\Sigma} = \mathbf{I}_p$.

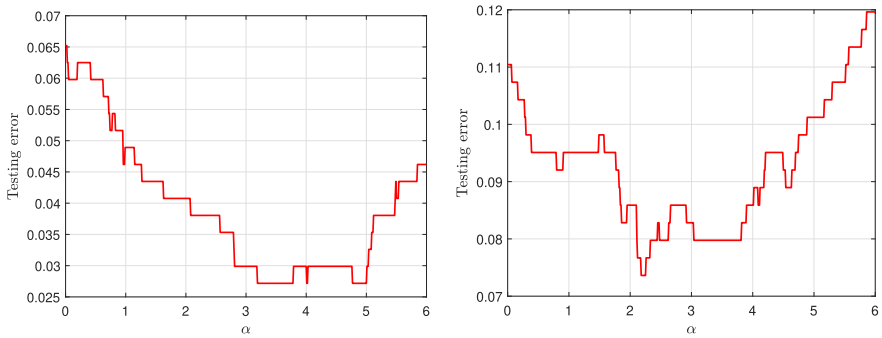
As the previous section shows, α-LDA exhibits the greatest improvement in performance among the sampled classifiers, particularly when the data dimensionality p is on the order of the number of samples n. This can be attributed to the fact that the LDA weight vector is an explicit function of the sample statistics. Due to estimation noise, there is much to be gained in this regime. We thus pursue an asymptotic study of α-LDA in growth regime where n and p grow at constant rates to each other. Under this growth regime, we derive an asymptotic expression and an estimator for the probability of misclassification of α-LDA. Note that this analysis is based on Gaussian data assumptions under both common and distinct class covariances. This is required in order to be able to derive exact expressions of the probability of misclassification for which the limit and estimator are computed.

A. ASYMPTOTIC ANALYSIS

In this section we first show that under the following growth regime assumptions

a) $0 < \liminf \frac{p}{n} < \limsup \frac{p}{n} < 1$
b) $\frac{n_i}{n} \to c_i \in (0, 1),\ i = 0, 1$
c) $\limsup_p \|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\|_2 < \infty$
d) $\limsup_p \|\boldsymbol{\Sigma}_i\|_2 < \infty,\ i = 0, 1$
e) $\liminf_p \lambda_{\min}(\boldsymbol{\Sigma}_i) > 0,\ i = 0, 1$

and considering the training set to be random, the probability of misclassification of the α-LDA classifier converges to a quantity that is a function of only true statistics. This quantity is referred to as the *deterministic equivalent* (DE) of the probability of misclassification. The DE approximates the random realization of the probability of misclassification, and can be useful for understanding the behavior of the classifier with synthetic data, for which the statistics are perfectly known. In practice, however, the statistics are unknown. For this reason, we also derive an estimator of the probability of



(a) USPS 2 and 6 (b) USPS 3 and 5

FIGURE. 8. Plots of testing error on USPS digit pairs of α-SVM with penalty set non-optimally.

misclassification which is consistent under the same growth assumptions. This is referred to as a *G-estimator* of the probability of misclassification and can be used to tune α. To proceed with these derivations, we first require an expression for the expected probability of misclassification.

Assuming the classes $\mathcal{C}_0$ and $\mathcal{C}_1$ are Gaussian with means and covariances $\boldsymbol{\mu}_0$, $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_1$ respectively, the probability of misclassification of a test point $\mathbf{x}$ by the α-LDA classifier has the form

$$\varepsilon = \pi_0 \Phi\left(\frac{m_0}{\sqrt{\sigma_0^2}}\right) + \pi_1 \Phi\left(-\frac{m_1}{\sqrt{\sigma_1^2}}\right)$$

where m_0, m_1, σ_0^2, and σ_1^2 are the discriminant means and variances conditioned on $\mathbf{x} \in \mathcal{C}_0$ and $\mathbf{x} \in \mathcal{C}_1$ respectively. Define $\rho = \frac{\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}}{\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}}}$. Then for $i = 0, 1$,

$$m_i = \left(\rho \hat{\boldsymbol{\mu}}^T + \alpha \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{P}_{\hat{\boldsymbol{\mu}}}\right)\left(\boldsymbol{\mu}_i - \frac{\hat{\boldsymbol{\mu}}_0 + \hat{\boldsymbol{\mu}}_1}{2}\right)$$

and

$$\sigma_i^2 = \left(\rho \hat{\boldsymbol{\mu}}^T + \alpha \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{P}_{\hat{\boldsymbol{\mu}}}\right) \boldsymbol{\Sigma}_i \left(\rho \hat{\boldsymbol{\mu}}^T + \alpha \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{P}_{\hat{\boldsymbol{\mu}}}\right)^T.$$

In the following sections, we present the DEs and G-estimators for both the general case of distinct covariances and the special case of common covariances.

1) DE OF THE PROBABILITY OF MISCLASSIFICATION

Formally, the DE of ε, denoted by $\bar{\varepsilon}$, is a sequence of p and n satisfying $\varepsilon - \bar{\varepsilon} \xrightarrow{\text{a.s.}} 0$ under the growth regime assumptions (a)-(f). For sequences $\bar{m}_i$ and $\bar{\sigma}_i^2$, $i = 0, 1$, such that

$$\begin{aligned} m_i - \bar{m}_i &\xrightarrow{\text{a.s.}} 0 \\ \sigma_i^2 - \bar{\sigma}_i^2 &\xrightarrow{\text{a.s.}} 0 \end{aligned} \tag{16}$$

under the growth regime assumptions (a)-(e), it is

$$\bar{\varepsilon} = \pi_0 \Phi\left(\frac{\bar{m}_0}{\sqrt{\bar{\sigma}_0^2}}\right) + \pi_1 \Phi\left(-\frac{\bar{m}_1}{\sqrt{\bar{\sigma}_1^2}}\right)$$

(see Lemma 2 in [22] for proof). Thus, the DE $\bar{\varepsilon}$ is itself a function of DEs $\bar{m}_0$, $\bar{m}_1$, $\bar{\sigma}_0^2$, and $\bar{\sigma}_1^2$ which are also functions of only true statistics.

In the following theorem, we state the expressions of $\bar{m}_0$, $\bar{m}_1$, $\bar{\sigma}_0^2$, and $\bar{\sigma}_1^2$ which are used to compute $\bar{\varepsilon}$. This is followed by a corollary which corresponds to the special case when $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$.[1] First, define the following quantities for $i = 0, 1$ and $j, k = 1, 2$,

$$\begin{aligned}
\bar{\mathbf{Q}} &= \left(\frac{n_0 - 1}{n-2}\frac{1}{1+\tilde{\delta}}\boldsymbol{\Sigma}_0 + \frac{n_1 - 1}{n-2}\frac{1}{1+\tilde{\nu}}\boldsymbol{\Sigma}_1\right)^{-1}, \\
\mathbf{A}_i &= \boldsymbol{\Sigma}_i \bar{\mathbf{Q}}, \\
R_{jk} &= \frac{n_{j-1} - 1}{n_{k-1} - 1}\left[(\mathbf{I}_2 - \boldsymbol{\Omega})^{-1}\boldsymbol{\Omega}\right]_{j,k}, \\
[\boldsymbol{\Omega}]_{1j} &= \frac{n_{j-1} - 1}{n-2}\left(\frac{1}{1+\tilde{\delta}}\right)^2 \frac{1}{n-2}\mathrm{tr}\mathbf{A}_0\mathbf{A}_{j-1}, \\
[\boldsymbol{\Omega}]_{2j} &= \frac{n_{j-1} - 1}{n-2}\left(\frac{1}{1+\tilde{\nu}}\right)^2 \frac{1}{n-2}\mathrm{tr}\mathbf{A}_1\mathbf{A}_{j-1}, \\
\tilde{\mathbf{Q}}_i &= \bar{\mathbf{Q}}\left(\mathbf{A}_i + R_{1(i+1)}\mathbf{A}_0 + R_{2(i+1)}\mathbf{A}_1\right), \\
\kappa &= \frac{\boldsymbol{\mu}^T \bar{\mathbf{Q}} \boldsymbol{\mu} + \frac{1}{n_0}\mathrm{tr}\mathbf{A}_0 + \frac{1}{n_1}\mathrm{tr}\mathbf{A}_1}{\boldsymbol{\mu}^T \boldsymbol{\mu} + \frac{1}{n_0}\mathrm{tr}\boldsymbol{\Sigma}_0 + \frac{1}{n_1}\mathrm{tr}\boldsymbol{\Sigma}_1}, \\
\eta &= \frac{\left(\frac{1}{1-\frac{p}{n-2}}\right)\left[\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{p}{n_0} + \frac{p}{n_1}\right]}{\boldsymbol{\mu}^T \boldsymbol{\mu} + \left(\frac{1}{n_0} + \frac{1}{n_1}\right)\mathrm{tr}\boldsymbol{\Sigma}}, \\
\tau &= \frac{1}{1 - \frac{p}{n-2}}, \ \bar{\alpha} = 1 - \alpha,
\end{aligned}$$

and $\tilde{\delta}$ and $\tilde{\nu}$ are the results of the fixed point iteration of $\tilde{\delta} = \frac{1}{n-2}\mathrm{tr}\boldsymbol{\Sigma}_0\bar{\mathbf{Q}}$ and $\tilde{\nu} = \frac{1}{n-2}\mathrm{tr}\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}$ for any positive initialization of $\tilde{\delta}$ and $\tilde{\nu}$.

[1] Note in these statements that while technically $n-2$ is equivalent to n asymptotically, we retain the $n-2$ in these expressions for increased accuracy in finite dimensions.

Theorem 2. (Distinct covariance DEs): The DEs $\bar{m}_0$, $\bar{m}_1$, $\bar{\sigma}_0^2$, and $\bar{\sigma}_1^2$ satisfying (16) under the growth regime assumptions (a)–(e) are given by

$$\bar{m}_i = \bar{\alpha}\kappa\left[\frac{(-1)^{i+1}}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} + \frac{1}{2}\left(\frac{1}{n_0}\mathrm{tr}\boldsymbol{\Sigma}_0 - \frac{1}{n_1}\mathrm{tr}\boldsymbol{\Sigma}_1\right)\right] + \alpha\left[\frac{(-1)^{i+1}}{2}\boldsymbol{\mu}^T\bar{\mathbf{Q}}\boldsymbol{\mu} + \frac{1}{2}\left(\frac{1}{n_0}\mathrm{tr}\mathbf{A}_0 - \frac{1}{n_1}\mathrm{tr}\mathbf{A}_1\right)\right]$$

and

$$\bar{\sigma}_i^2 = \bar{\alpha}^2\kappa^2\left[\boldsymbol{\mu}^T\boldsymbol{\Sigma}_i\boldsymbol{\mu} + \frac{1}{n_0}\mathrm{tr}\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_i + \frac{1}{n_1}\mathrm{tr}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_i\right] + 2\alpha\bar{\alpha}\kappa\left[\boldsymbol{\mu}^T\mathbf{A}_i\boldsymbol{\mu} + \frac{1}{n_0}\mathrm{tr}\boldsymbol{\Sigma}_i\mathbf{A}_0 + \frac{1}{n_1}\mathrm{tr}\boldsymbol{\Sigma}_i\mathbf{A}_1\right] + \alpha^2\left[\boldsymbol{\mu}^T\tilde{\mathbf{Q}}_i\boldsymbol{\mu} + \frac{1}{n_0}\mathrm{tr}\boldsymbol{\Sigma}_0\tilde{\mathbf{Q}}_i + \frac{1}{n_1}\mathrm{tr}\boldsymbol{\Sigma}_1\tilde{\mathbf{Q}}_i\right]$$

for $i = 0, 1$.

Proof: See Appendix B-A of our extended report [1].

Corollary 2. (Common covariance DEs): The DEs $\bar{m}_0$, $\bar{m}_1$, $\bar{\sigma}_0^2$, and $\bar{\sigma}_1^2$ satisfying (16) under the growth regime assumptions (a)–(e) are given by

$$\bar{m}_i = \bar{\alpha}\eta\left(\frac{(-1)^{i+1}}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} + \frac{1}{2}\left(\frac{1}{n_0} - \frac{1}{n_1}\right)\mathrm{tr}\boldsymbol{\Sigma}\right) + \alpha\left[\frac{\tau}{2}\left[(-1)^{i+1}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{p}{n_0} - \frac{p}{n_1}\right]\right]$$

and

$$\bar{\sigma}_i^2 = \bar{\alpha}^2\eta^2\left[\boldsymbol{\mu}^T\boldsymbol{\Sigma}\boldsymbol{\mu} + \left(\frac{1}{n_0} + \frac{1}{n_1}\right)\mathrm{tr}\boldsymbol{\Sigma}^2\right] + \alpha^2\tau^3\left[\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{p}{n_0} + \frac{p}{n_1}\right] + 2\alpha\bar{\alpha}\tau\eta\left[\boldsymbol{\mu}^T\boldsymbol{\mu} + \left(\frac{1}{n_0} + \frac{1}{n_1}\right)\mathrm{tr}\boldsymbol{\Sigma}\right]$$

for $i = 0, 1$.

Proof: See Appendix B-B of our extended report [1].

2) G-ESTIMATOR OF THE PROBABILITY OF MISCLASSIFICATION

The G-estimator $\hat{\varepsilon}$ of the probability of misclassification ε is a function of sample statistics $\hat{\boldsymbol{\mu}}_0$, $\hat{\boldsymbol{\mu}}_1$, $\hat{\boldsymbol{\Sigma}}_0$, and $\hat{\boldsymbol{\Sigma}}_1$ such that $\hat{\varepsilon} - \varepsilon \xrightarrow{\text{a.s.}} 0$ under the growth regime assumptions (a)-(f). For sequences $\hat{m}_i$ and $\hat{\sigma}_i^2$, $i = 0, 1$, which are functions of only sample statistics, such that

$$\hat{m}_i - m_i \xrightarrow{\text{a.s.}} 0$$
$$\hat{\sigma}_i^2 - \sigma i^2 \xrightarrow{\text{a.s.}} 0 \tag{17}$$

under the growth regime assumptions (a)-(e), it is

$$\hat{\varepsilon} = \hat{\pi}_0\Phi\left(\frac{\hat{m}_0}{\sqrt{\hat{\sigma}_0^2}}\right) + \hat{\pi}_1\Phi\left(-\frac{\hat{m}_1}{\sqrt{\hat{\sigma}_1^2}}\right).$$

The following theorem states the expressions of $\hat{m}_0$, $\hat{m}_1$, $\hat{\sigma}_0^2$, and $\hat{\sigma}_1^2$ which are used to compute $\hat{\varepsilon}$. This is followed by a corollary which is specific to the case when $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$ is assumed. First, define

$$\lambda_i = \frac{\frac{1}{n-2}\mathrm{tr}\hat{\boldsymbol{\Sigma}}_i\hat{\boldsymbol{\Sigma}}^{-1}}{1 - \frac{1}{n-2}\mathrm{tr}\hat{\boldsymbol{\Sigma}}_i\hat{\boldsymbol{\Sigma}}^{-1}}.$$

Theorem 3. (Distinct covariance G-estimators): The G-estimators $\hat{m}_0$, $\hat{m}_1$, $\hat{\sigma}_0^2$, and $\hat{\sigma}_1^2$, satisfying (17) under the growth regime assumptions (a)-(e) are given by

$$\hat{m}_i = (-1)^{i+1}\bar{\alpha}\rho\left(\frac{1}{2}\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}} - \frac{1}{n_i}\mathrm{tr}\hat{\boldsymbol{\Sigma}}_i\right) + (-1)^{i+1}\alpha\left(\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}} - \frac{n-2}{n_i}\lambda_i\right)$$

and

$$\hat{\sigma}_i^2 = (1-\alpha)^2\rho^2\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\Sigma}}_i\hat{\boldsymbol{\mu}} + 2\alpha\bar{\alpha}\rho\,(1+\lambda_i)\,\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\Sigma}}_i\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}} + \alpha^2\,(1+\lambda_i)^2\,\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\Sigma}}_i\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}$$

for $i = 0, 1$.

Proof: See Appendix C-A of our extended report [1].

Corollary 3. (Common covariance G-estimators): The G-estimators $\hat{m}_0$, $\hat{m}_1$, $\hat{\sigma}_0^2$, and $\hat{\sigma}_1^2$, satisfying (17) under the growth regime assumptions (a)-(e) are given by

$$\hat{m}_i = \frac{(-1)^{i+1}}{2}\left(\rho\hat{\boldsymbol{\mu}}^T + \alpha\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{P}_{\hat{\boldsymbol{\mu}}}\right)\hat{\boldsymbol{\mu}} + (-1)^{i+1}\left[\rho(\alpha-1)\frac{1}{n_i}\mathrm{tr}\hat{\boldsymbol{\Sigma}} - \alpha\frac{\frac{p}{n_i}}{1 - \frac{p}{n-2}}\right]$$

and

$$\hat{\sigma}_i^2 = \rho^2\bar{\alpha}^2\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\mu}} + \alpha^2\tau^2\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}} + 2\alpha\rho\bar{\alpha}\tau\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}$$

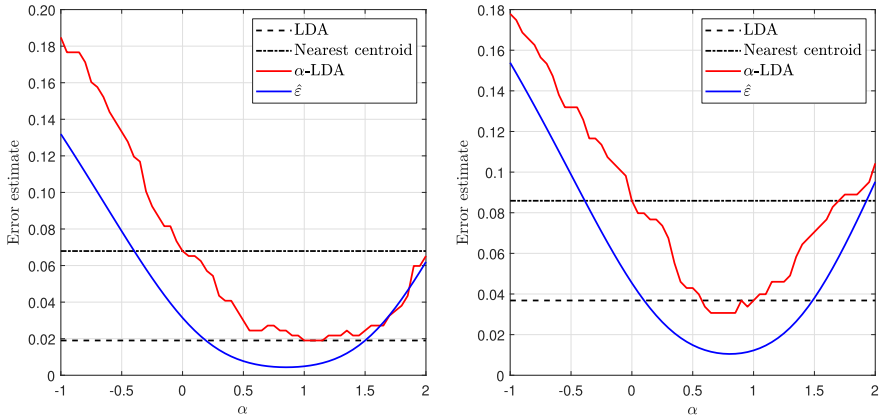
for $i = 0, 1$.

Proof: See Appendix C-B of our extended report [1].

Notice that $\hat{\varepsilon}$ is a function of the sample statistics. It estimates the probability of misclassification without the need for additional testing data and it is much more computationally efficient than the cross-validation procedure. In the next section, we show how to use $\hat{\varepsilon}$ for the purpose of tuning the α parameter.

B. TUNING THE α-LDA PARAMETER

In this section, α-LDA is applied to real data. The objective is to show how α-LDA performs as compared to LDA and the nearest centroid classifier on real data, as well as to demonstrate the use of the G-estimator $\hat{\varepsilon}$ in tuning the α parameter. We consider binary classification of digit pairs from the USPS dataset [23] and phoneme pairs from the dataset [24]. For each problem, we train and test LDA, nearest centroid, and



(a) USPS 2 and 6 (b) USPS 5 and 8

FIGURE. 9. Plots of testing error estimates of classifying USPS digit pairs for LDA, the nearest centroid, and α-LDA as well as the G-estimator $\hat{\varepsilon}$ of the α-LDA expected testing error.

α-LDA on the relevant dataset. The empirical errors are plotted against varying α. Also plotted is the G-estimator $\hat{\varepsilon}$ of the error of α-LDA.[2]

Fig. 9 shows the results on two digit pairs from the USPS dataset. As mentioned in Section II-B1, this dataset consists of grayscale images of handwritten digits $0-9$ encoded as 256-dimensional vectors.

For Fig. 9(a), we use the digit pair '2' and '6'. Overall, there are $n = 1395$ total training vectors and 368 total testing vectors corresponding to this digit pair. The figure shows that LDA achieves the lowest empirical error on this digit pair. This performance is matched by α-LDA at $\alpha = 1$. Although $\hat{\varepsilon}$ does not exactly match the empirical error, for parameter tuning it suffices that it follows the same trend. In this case, if we had directly used $\hat{\varepsilon}$ to tune the α parameter, we would have set it to $\alpha = 0.85$. This setting results in an increase of merely 0.0054 in error compared to the optimal setting. For more sensitive applications, the parameter setting suggested by the G-estimator may be used as a starting point from which to search for the optimal α using a more accurate (but computationally-intensive) method.

For Fig. 9(b), we use the digit pair '5' and '8'. Overall, there are $n = 1098$ total training vectors and 326 total testing vectors corresponding to this digit pair. In this case, α-LDA achieves the lowest error of 0.0307 at $\alpha = 0.65$. This is a 16.6% decrease in error relative to LDA which has an error rate of 0.0368. If we had directly used $\hat{\varepsilon}$ to tune the α parameter, we would have set it to $\alpha = 0.8$. This setting incurs no loss in accuracy. Notice this dataset has less training samples than the last one. The increased estimation noise explains why α-LDA is able to provide a performance advantage over LDA.

Fig. 10 considers a phoneme pair. The phoneme dataset consists of a total of 4509 instances of digitized speech vectors

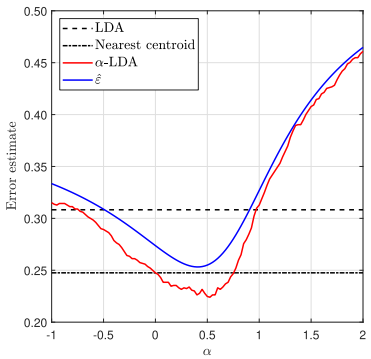


FIGURE. 10. Plots of testing error estimates of classifying phonemes 'aa' and 'ao' for LDA, the nearest centroid, and α-LDA as well as the G-estimator $\hat{\varepsilon}$ of the α-LDA expected testing error.

of the five phonemes 'aa,' 'ao,' 'dcl,' 'iy,' and 'sh,' having 256 features each. All 1717 instances of the phonemes 'ao' and 'aa' (which are the closest in pronunciation) were extracted in order to construct this binary classification problem. As the dataset is not pre-divided into training and testing sets, the splitting was performed randomly. We take advantage of this to construct a classification problem in which n is not much greater than p. A training set consisting of 400 samples is randomly extracted from the full set of 'aa' and 'ao' phonemes according to the same proportions. This leaves 1317 samples for testing. Based on the simulations from the previous section, we expect to observe a much greater performance gain in this scenario compared to Fig. 9.

Fig. 10 shows that, as expected, α-LDA significantly outperforms LDA with an error of 0.224 corresponding to the former compared to 0.3083 corresponding to the latter. It achieves a 27.3% decrease in error at $\alpha = 0.525$. In this case, it seems that the data leans more towards an isotropic covariance structure, as nearest centroid performs better than LDA. Even so, $\alpha = 0$ is not optimal. Thus, α-LDA provides the best balance between both of these classifiers. Lastly, the

[2]Note that for these particular datasets, the two G-estimators almost match. Out of the two, the G-estimator which assumes common covariances is plotted.

G-estimator points towards an α setting of 0.4. Using this setting incurs an increase in error of just 0.0023 relative to the optimal setting.

IV. CONCLUSION

In this work, we design a method of weight vector tuning for binary linear classifiers based on the decomposition of the discriminant into informative and noisy components. The tuning takes the form of a linear parameterization of the decomposition. Deriving this method reveals a novel interpretation of the classic LDA classifier weight vector as minimizing the noise from the test point to which it is applied.

We simulate the performance gain of this method for a variety of linear classifiers: LDA, SVM, logistic regression, R-LDA, and RP-LDA ensemble, and under different data dimensionality and sample size settings. Firstly, we find that weight vector tuning can compensate performance loss due to poorly chosen native classifier hyperparameters. It thus eliminates the need for native hyperparameter tuning. As weight vector tuning occurs post weight vector generation, this can be advantageous in terms of computational efficiency when the native hyperparameters need to be set prior to weight vector generation. Secondly, we find that the parameterization significantly improves the performance of LDA under high estimation noise. We proceed to derive the parameterized LDA classifier misclassification probability in the RMT growth regime corresponding to these settings, in which the data dimensionality and sample size grow at comparable rates to each other. We also provide an estimator of the probability of misclassification which neither relies on additional data samples nor requires intensive computations, and thus can be used to tune the parameter of this classifier in a computationally efficient manner.

REFERENCES

[1] L. B. Niyazi, A. Kammoun, H. Dahrouj, M.-S. Alouini, and T. Al-Naffouri, "Weight vector tuning and asymptotic analysis of binary linear classifiers," 2021. [Online]. Available: https://arxiv.org/abs/2110.00567

[2] A. F. Marquand and S. M. Kia, "Chapter 5 - Linear methods for classification," in *Machine Learning*, A. Mechelli and S. Vieira, Eds., Cambridge, MA, USA: Academic Press, 2020, pp. 83–100. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128157398000055

[3] J. Wen et al., "Performance of linear classification algorithms on α/γ discrimination for LaBr3: Ce scintillation detectors with various pulse digitizer properties," *J. Instrum.*, vol. 15, no. 2, 2020, Art. no. P02004.

[4] D. P. Berry, T. Pabiou, R. Fanning, R. D. Evans, and M. M. Judge, "Linear classification scores in beef cattle as predictors of genetic merit for individual carcass primal cut yields," *J. Animal Sci.*, vol. 97, no. 6, pp. 2329–2341, 2019.

[5] G. S. Randhawa, M. P. Soltysiak, H. El Roz, C. P. de Souza, K. A. Hill, and L. Kari, "Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study," *Plos one*, vol. 15, no. 4, 2020, Art. no. e0232391.

[6] P. Ashok, T. Brázdil, K. Chatterjee, J. Křetínskỳ, C. H. Lampert, and V. Toman, "Strategy representation by decision trees with linear classifiers," in *Proc. Int. Conf. Quantitative Eval. Syst.*, 2019, pp. 109–128.

[7] R. Alanni, J. Hou, H. Azzawi, and Y. Xiang, "A novel gene selection algorithm for cancer classification using microarray datasets," *BMC Med. Genomic.*, vol. 12, no. 1, 2019, Art. no. 10.

[8] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, "Recent advances of large-scale linear classification," *Proc. IEEE*, vol. 100, no. 9, pp. 2584–2603, Sep. 2012.

[9] R. O. Duda, D. G. Stork, and P. E. Hart, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2001.

[10] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Statistics Series). New York, NJ, USA: Springer, 2001.

[11] C. Wang and B. Jiang, "On the dimension effect of regularized linear discriminant analysis," *Electron. J. Statist.*, vol. 12, no. 2, pp. 2709–2742, 2018.

[12] A. Zollanvari, M. Abdirash, A. Dadlani, and B. Abibullaev, "Asymptotically bias-corrected regularized linear discriminant analysis for cost-sensitive binary classification," *IEEE Signal Process. Lett.*, vol. 26, no. 9, pp. 1300–1304, Sep. 2019.

[13] S. Huang, T. Tong, and H. Zhao, "Bias-corrected diagonal discriminant rules for high-dimensional classification," *Biometrics*, vol. 66, no. 4, pp. 1096–1106, 2010.

[14] H. Sifaou, A. Kammoun, and M.-S. Alouini, "High-dimensional linear discriminant analysis classifier for spiked covariance model," *J. Mach. Learn. Res.*, vol. 21, pp. 1–24, 2020.

[15] Q. Mai, H. Zou, and M. Yuan, "A direct approach to sparse discriminant analysis in ultra-high dimensions," *Biometrika*, vol. 99, no. 1, pp. 29–42, 2012.

[16] L. B. Niyazi, A. Kammoun, H. Dahrouj, M.-S. Alouini, and T. Y. Al-Naffouri, "Asymptotic analysis of an ensemble of randomly projected linear discriminants," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 3, pp. 914–930, Nov. 2020. [Online]. Available: https://arxiv.org/abs/2004.08217

[17] E. W. Weisstein, "Hypersphere point picking," *From MathWorld–A Wolfram Web Resource*, 2017. [Online]. Available: http://mathworld.wolfram.com/HyperspherePointPicking.html

[18] Y. Guo, T. Hastie, and R. Tibshirani, "Regularized linear discriminant analysis and its application in microarrays," *Biostatistics*, vol. 8, no. 1, pp. 86–100, 2007.

[19] R. J. Durrant and A. Kabán, "Random projections as regularizers: Learning a linear discriminant ensemble from fewer observations than dimensions," in *Proc. Asian Conf. Mach. Learn.*, 2013, vol. 29, 2013, pp. 17–32. [Online]. Available: http://jmlr.org/proceedings/papers/v29/Durrant13.htmlGoogleScholar

[20] L. Bottou and C.-J. Lin, "Support vector machine solvers," *Large Scale Kernel Machines*, vol. 3, no. 1, pp. 301–320, 2007.

[21] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4. New York, NY, USA: Springer, 2006.

[22] L. B. Niyazi, A. Kammoun, H. Dahrouj, M.-S. Alouini, and T. Y. Al-Naffouri, "Asymptotic analysis of an ensemble of randomly projected linear discriminants," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 3, pp. 914–930, Nov. 2020.

[23] Y. Le Cun et al., "Handwritten zip code recognition with multilayer networks," in *Proc. 10th Int. Conf. Pattern Recognit.*, 1990, vol. 2, pp. 35–40.

[24] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," *Ann. Statist.*, vol. 23, no. 1, pp. 73–102, 1995.

LAMA B. NIYAZI received the B.Sc. degree in electrical and computer engineering from Effat University, Jeddah, Saudi Arabia, in 2015, and the M.Sc degree in electrical engineering in 2017 from the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, where she is currently working toward the Ph.D. degree with the Electrical and Computer Engineering Program. Her research focuses on the application of random matrix theory to machine learning.

ABLA KAMMOUN (Member, IEEE) was born in Sfax, Tunisia. She received the Engineering degree in signal and systems from Tunisia Polytechnic School, La Marsa, Tunisia, and the masters and Ph.D. degrees in digital communications from Telecom Paris Tech, Paris, France (formerly, Ecole Nationale Supeure des T). From 2010 to 2012, she was a Postdoctoral Researcher with the TSI Department, Telecom ParisTech. She was with Supelec, Alcatel-Lucent Chair on Flexible Radio until 2013. She is currently a Research Scientist with the King Abdullah University of Science and Technology, Saudi Arabia. Her research interests include performance analysis of wireless communication systems, random matrix theory, and statistical signal processing.

HAYSSAM DAHROUJ (Senior Member, IEEE) received the B.E. degree (with high distinction) in computer and communications engineering from the American University of Beirut, Beirut, Lebanon, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Toronto (UofT), Toronto, ON, Canada, in 2010. In July 2020, he joined the Center of Excellence for NEOM Research, King Abdullah University of Science and Technology (KAUST), Saudi Arabia as a Senior Research Scientist. From June 2015 to June 2020, he was with the Department of Electrical and Computer Engineering, Effat University, Jeddah, Saudi Arabia, as an Assistant Professor, and as a Visiting Scholar with the Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) division at KAUST, where he also was a Research Associate between April 2014 and May 2015. Prior to joining KAUST, he was an Industrial Postdoctoral Fellow, UofT, in collaboration with BLiNQ Networks Inc., Kanata, Canada, where he worked on developing practical solutions for the design of non-line-of sight wireless backhaul networks. His contributions to the field led to five patents. During his doctoral studies at UofT, he pioneered the idea of coordinated beamforming as a means of minimizing intercell interference across multiple base stations. His main research interests include 6G wireless systems, cloud- and fog-radio access networks, cross-layer optimization, cooperative networks, convex optimization, machine learning, distributed algorithms, and optical communications. The journal paper on this subject was ranked second in the 2013 IEEE Marconi paper awards in wireless communications. Dr. Dahrouj was the recipient of both the Faculty Award of excellence in research, and the Faculty Award of excellence in teaching (at the university level) in May 2017. He is an Associate Editor for the *Frontiers in Communications and Networks*, and a Lead-Guest Editor of the Frontiers special issue on Resource Allocation in Cloud-Radio Access Networks and Fog-Radio Access Networks for B5G Systems.

MOHAMED-SLIM ALOUINI (Fellow, IEEE) was born in Tunis, Tunisia. He received the Ph.D. degree in electrical engineering from the California Institute of Technology (Caltech), Pasadena, CA, USA, in 1998. He was a Faculty Member with the University of Minnesota, Minneapolis, MN, USA. Then, he was with Texas A&M University at Qatar, Education City, Doha, Qatar, before joining the King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, as a Professor of electrical engineering in 2009. His research interests include modeling, design, and performance analysis of wireless communication systems.

TAREQ Y. AL-NAFFOURI (Senior Member, IEEE) received the B.S. degrees in mathematics and electrical engineering (with first Hons.) from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, the M.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, Georgia, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2004. He was a Visiting Scholar with the California Institute of Technology, Pasadena, CA, in 2005 and summer 2006. He was a Fulbright Scholar with the University of Southern California, Los Angeles, CA, in 2008. He is currently a Professor with Electrical Engineering Department, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. His research interests include sparse, adaptive, and statistical signal processing and their applications to wireless communications and localization, machine learning, and network information theory. He has more than 350 publications in journal and conference proceedings and 24 issued/pending patents. Dr. Al-Naffouri was the recipient of the IEEE Education Society Chapter Achievement Award in 2008 and Al-Marai Award for innovative research in communication in 2009. Dr. Al-Naffouri was an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING during 2013–2018.