Information Discriminant Analysis: Feature Extraction with an Information-Theoretic Objective

Zoran Nenadic, *Member*, *IEEE*

Abstract—Using elementary information-theoretic tools, we develop a novel technique for linear transformation from the space of observations into a low-dimensional (feature) subspace for the purpose of classification. The technique is based on a numerical optimization of an information-theoretic objective function, which can be computed analytically. The advantages of the proposed method over several other techniques are discussed and the conditions under which the method reduces to linear discriminant analysis are given. We show that the novel objective function enjoys many of the properties of the mutual information and the Bayes error and we give sufficient conditions for the method to be Bayes-optimal. Since the objective function is maximized numerically, we show how the calculations can be accelerated to yield feasible solutions. The performance of the method compares favorably to other linear discriminant-based feature extraction methods on a number of simulated and real-world data sets.

Index Terms—Feature extraction, information theory, mutual information, entropy, classification, linear discriminant analysis, Bayes error.

1 Introduction

Feature extraction is a common preprocessing step in the analysis of multivariate statistical data. In a broad sense, the feature extraction can be defined as a low-dimensional data representation, where features capture some important data properties. An obvious benefit of this dimensionality reduction is that data becomes computationally more manageable. More importantly, the dimensionality reduction facilitates numerous applications, such as data compression, denoising, pattern recognition, etc. Many of these applications rely on accurate estimates of the data statistics, e.g., means, covariances, or probability density functions (PDFs), in general. In the face of a limited sample size, a direct estimation of these quantities in the data space is grossly inaccurate, primarily because the high-dimensional data space is mostly sparse [1, p. 70], [2], a phenomenon known as the *curse of dimensionality*. Also, the feature extraction may be useful for visualization purposes, where the optimal low-dimensional data projection is sought (*projection pursuit*), subject to a suitably chosen objective function [2], [3].

In the majority of applications, feature extraction methods are linear; that is, a feature vector represents a linear combination of the attributes of a data vector and resides in a (linear) subspace of the data space. Consequently, the transformation from the data space to the feature space is represented by a transformation matrix. Recently, a couple of nonlinear feature extraction methods have been proposed [4], [5], where features reside on a low-dimensional manifold embedded in the data space. However, linear feature extraction methods continue to play an important role in many applications, primarily due to their computational effectiveness.

From a different standpoint, feature extraction methods can be classified as unsupervised and supervised. In unsupervised applications, the data class labels are unknown; therefore, it is assumed that data is sampled from a common distribution. The statistical properties of this distribution are then used to facilitate the feature extraction process. The best-known representative of these techniques is *principal component analysis* (PCA), which relies on the first two statistical moments of the data distribution (see [6] or any other textbook). Another popular method, which utilizes higher order statistical moments and can be viewed as a generalization of PCA, is *independent component analysis* (ICA) (see [7] for survey). In supervised applications, the knowledge of the data class labels is used to find a low-dimensional representation which preserves the class differences, so that a classifier can be designed in the feature domain. This type of feature extraction is often referred to as the discriminant feature extraction (DFE). Undoubtedly, *linear discriminant analysis* (LDA) is the best known representative of these techniques.

The basic form of LDA was introduced by Fisher [8] and, so, the method often goes by the name of the Fisher linear discriminant. The method was subsequently generalized by Rao [9] and it has since been used in many statistical applications, such as speech recognition (see [10] for a review), document classification [11], and face recognition [12]. Under fairly restrictive assumptions, it can be shown that LDA is an optimal[1] linear DFE [13]. Motivated by these restrictions, Kumar and Andreou [10] proposed an extension of LDA, which they derived using a maximum likelihood (ML) approach. Other linear DFE methods include the use of *probabilistic distance measures* [14], such as Kullback-Liebler

- *The author is with the Department of Biomedical Engineering and the Department of Electrical Engineering and Computer Science, University of California, Irvine, 3120 Natural Sciences II, Irvine, CA 92697-2715. E-mail: znenadic@uci.edu.*

Manuscript received 13 Nov. 2005; revised 28 June 2006; accepted 22 Sept. 2006; published online 18 Jan. 2007.
Recommended for acceptance by X. Tang.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0617-1105.
Digital Object Identifier no. 10.1109/TPAMI.2007.1156.

[1] The optimality is in the sense of Bayes (see Section 2).

(KL) divergence [15], [16], Bhattacharyya distance [16], and approximate Chernoff distance [17]. It should be noted that these criteria are designed for binary classification tasks and that their application to multiclass problems requires heuristic extensions, e.g., the introduction of an average pairwise distance.

Devijver and Kittler [14] also discuss several *probabilistic dependence measures*, which are naturally defined in a multiclass case and can serve as class-separability measures. The best-known representative of these measures is the mutual information, whose use in pattern recognition applications has been mostly limited to feature selection[2] problems [18], [19] due to the high-computational cost associated with the mutual information evaluation. However, current feature selection approaches lack a consistent strategy for combining individual features into a feature set and proposed solutions (see [20] for a review) are either of unrestricted combinatorial complexity or likely to be suboptimal [21]. Principe et al. [22] used alternative definitions of entropy [23], coupled with a Parzen window density estimation that led to a computationally simpler definition of the mutual information. Motivated by these findings, Torkkola developed an information-theoretic feature extraction algorithm [24], although his method is computationally demanding and seems impractical even for moderately sized data.

In this paper, we propose a novel information-theoretic class-separability measure which, unlike the mutual information, can be found analytically. In Section 2, we give an overview of the classical tools such as Bayes classification and LDA. The shortcomings of LDA serve as a motivation for the development of our technique. Section 3 introduces the idea behind our method, conveniently called *information discriminant analysis* (IDA), and some interesting theoretical properties of IDA are presented. A practical recipe for calculating the IDA feature extraction matrix is given as well. The performance of our method on several benchmark data sets is tested in Section 4. Discussion is presented in Section 5 and concluding remarks are given in Section 6. Some mathematical derivations and proofs are given in the supplemental Appendix, which can be found at http://computer.org/tpami/archives.htm.

2 Bayes Classifier and Linear Discriminant Analysis

To objectively assess the performance of any DFE method, a classifier is designed in the feature domain. For theoretical purposes, the Bayes classifier is often used as a standard benchmark. Let $f_{\mathrm{R}|\Omega}(\boldsymbol{r} \mid \omega_i)$ be the PDF of a continuous random variable (RV) $\mathrm{R} \in \mathbb{R}^n$ conditioned upon a class variable $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$, where classes are drawn from a discrete distribution with the *prior* probability $p_i \overset{\triangle}{=} P(\Omega = \omega_i)$, $\forall i = 1, 2, \cdots, c$. The probability of misclassification is given by

$$P_{\mathrm{R}}(\varepsilon) = 1 - \sum_{i=1}^{c} \int_{\mathcal{R}_i} f_{\mathrm{R}|\Omega}(\boldsymbol{r} \mid \omega_i)\, p_i \, d\boldsymbol{r}, \tag{1}$$

where $\mathcal{R}_i \subset \mathbb{R}^n$ is the region of acceptance of the class ω_i, as determined by the classifier. A classifier that minimizes (1) is the Bayes classifier

$$i^* = \arg \max_{1 \le i \le c} P(\omega_i \mid \boldsymbol{r}_0), \tag{2}$$

where $\boldsymbol{r}_0$ is an unlabeled observation and $P(\omega_i \mid \boldsymbol{r}_0)$ is the *posterior* class probability. Consequently, the minimum $\varepsilon_{\mathrm{R}} \overset{\triangle}{=} \min P_{\mathrm{R}}(\varepsilon)$ is called the *Bayes error*. An important property of the Bayes error is that it is invariant under invertible linear transformations. Unfortunately, the evaluation of ε_{R} requires the knowledge of $f_{\mathrm{R}|\Omega}(\boldsymbol{r} \mid \omega_i)$, which limits the applicability of the Bayes error, in practice, where the class-conditional PDFs are rarely known and have to be estimated from data. If the sample size is relatively small with respect to the data dimension, n, these estimates may be quite unreliable. One remedy is to extract low-dimensional features by virtue of a full-rank[3] linear transformation $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^m$, where m $(m \ll n)$ is the dimension of the feature space. The classifier (2), or any other classifier, can then be designed for the features $\bar{\mathrm{R}} \overset{\triangle}{=} \boldsymbol{T}\,\mathrm{R}$. While theoretical analysis shows that the performance of the Bayes classifier can only deteriorate under such a transformation [25, p. 110], i.e., $\varepsilon_{\bar{\mathrm{R}}} \ge \varepsilon_{\mathrm{R}}$, the opposite effect is often seen in dealing with finite data samples (see Section 4.2).

Due to its computational simplicity, LDA is a widely used DFE technique. There are many variants of LDA and a somewhat unifying definition can be found in [1, p. 446]. In its most popular form, LDA maximizes the generalized Rayleigh quotient

$$J(\boldsymbol{T}) = \frac{|\boldsymbol{T}\, \Sigma_{\mathrm{B}} \boldsymbol{T}^{\mathrm{T}}|}{|\boldsymbol{T}\, \Sigma_{\mathrm{W}} \boldsymbol{T}^{\mathrm{T}}|},$$

where Σ_{B} and Σ_{W} represent the between-class and the within-class scatter matrices [25, p. 121], respectively, and $\boldsymbol{T}$ is the feature extraction matrix. The biggest appeal of LDA is that such a matrix can be found analytically, thereby avoiding numerical optimization.

Since the evaluation of Σ_{B} and Σ_{W} relies on the first two statistical moments, LDA implicitly assumes Gaussian classes. Additionally, if the class-conditional covariances are equal, it can be shown that LDA is optimal in the sense of Bayes [13, p. 90], i.e., $\varepsilon_{\bar{\mathrm{R}}} = \varepsilon_{\mathrm{R}}$, where $\bar{\mathrm{R}} = \boldsymbol{T}\,\mathrm{R}$ is the extracted feature vector. If the classes do not conform to these so-called *homoscedastic* conditions, LDA will be suboptimal, even if the classes *are* Gaussian. In general, if the class-conditional means are similar and/or the class-conditional covariances are different, LDA may be highly suboptimal. Fig. 1a shows one such example, where searching for the best 1D features yields quite different results for LDA and IDA (to be described in Section 3). In particular, the features extracted by LDA are fully overlapped, resulting in the Bayes error of 50 percent, compared to 25.78 percent error of the IDA features. The subspace extracted by a recently developed technique of Loog and Duin [17] has also been shown. Although the method maximizes a criterion based on the Chernoff distance [1, p. 98], it too fails to find a useful data projection; in particular, a subspace with zero Chernoff distance is extracted. Thus, we will refer to this technique as the

[2] We make a distinction between the feature selection and the feature extraction. The feature extraction gives rise to a feature vector by utilizing the joint statistical properties of features. The feature selection is concerned with individual (scalar) features, where the feature space is constructed by the concatenation of individual features.

[3] A deficient-rank transformation matrix would extract features that are linear combinations of other features. To avoid this redundancy, only full-rank extraction matrices will be considered in this article.

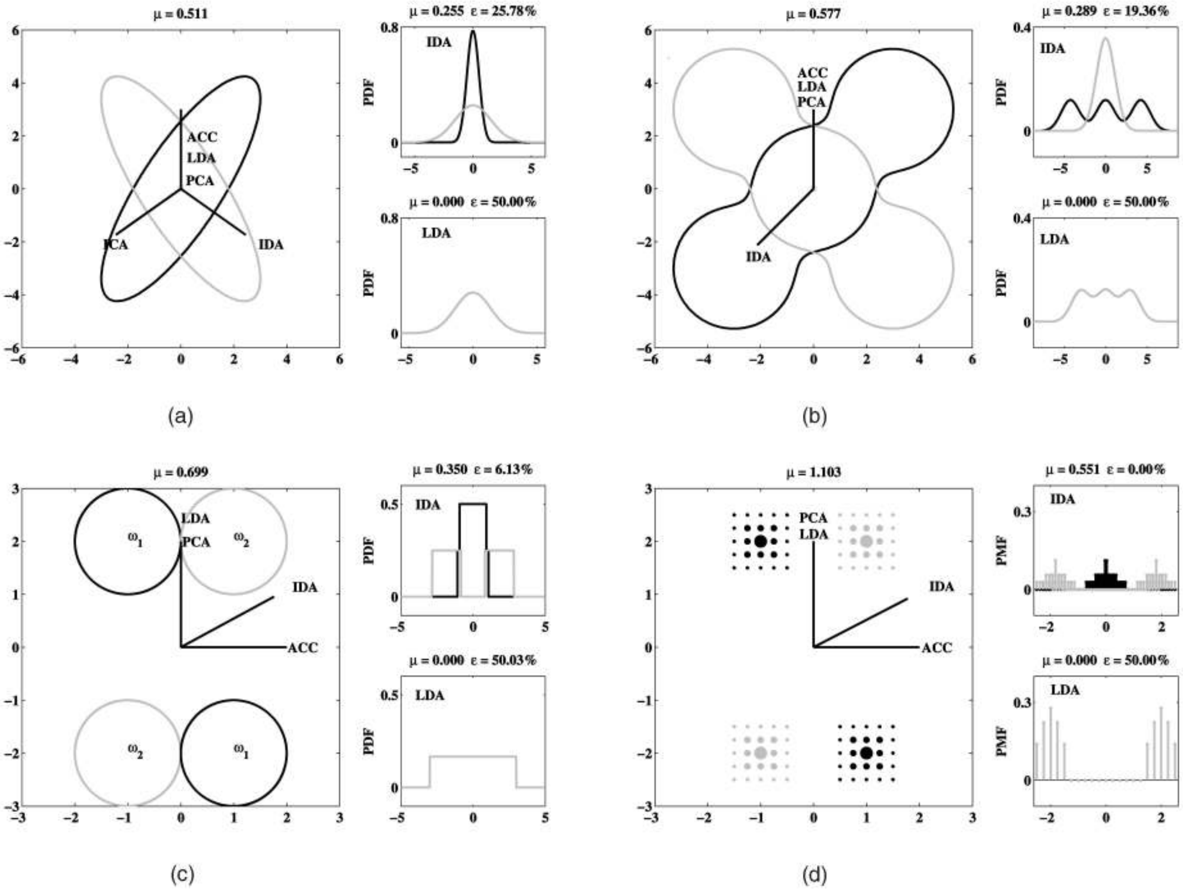

Fig. 1. Various 2D binary class examples where LDA fails. The class probabilities are uniform, i.e., $p_1 = p_2$. The straight lines indicate optimal 1D subspace according to different feature extraction methods: IDA, LDA, ACC, and PCA. The numbers above the panels represent the value of information theoretic objective μ (see Section 3.1), and the estimate ε of the Bayes error, evaluated through a numeric integration. The two smaller panels are the class-conditional PDFs of the extracted features. (a) Gaussian classes. (b) Mixture of Gaussian classes. (c) Bimodal uniform classes. (d) Discrete class distributions with the size of the dot proportional to the probability mass function (PMF).

approximate Chernoff criterion (ACC). Two unsupervised techniques, PCA and ICA, have been shown for reference. The direction of maximum variance reveals nothing about the two classes, and so the PCA features are useless. On the contrary, ICA produces essentially the same result as our method (note the symmetry of the example). We will underscore the similarities between the two methods later on. We emphasize, however, that the two techniques are fundamentally different (supervised versus unsupervised). Figs. 1b, 1c, and 1d show that, if the class-conditional means are overlapped, the superiority of IDA over LDA and ACC extends beyond Gaussian classes.

Another weakness of LDA is that the size of the feature space is constrained by the number of classes c. Since $\mathrm{rank}(\Sigma_\mathrm{B}) \leq c - 1$ and the LDA feature space is spanned by the eigenvectors of $\Sigma_\mathrm{W}^{-1}\,\Sigma_\mathrm{B}$ corresponding to its nonzero eigenvalues, the number of features is at most $c - 1$. While the Bayes classifier (2) indeed requires $c - 1$ features, namely, the posteriors $P(\omega_i \mid \boldsymbol{r})$, $i = 1, 2, \cdots, c$,[4] these features cannot be extracted from data in a linear fashion. Therefore, for linear feature extraction methods, there is no theoretical basis for a feature vector to be $c - 1$-dimensional.

Examples illustrating this shortcoming of LDA are presented in Table 1. Two Gaussian class-conditional PDFs with randomly sampled parameters were created in 2D, 3D, and 4D spaces and the optimal features were extracted using LDA, ACC, and IDA methods. The experiment was repeated 1,000 times with random sampling performed on a trial-by-trial basis. Since $c = 2$, LDA features are constrained to a 1D subspace, while IDA and ACC can search for discriminatory information in higher dimensions. For each feature space, the Bayes error was estimated using Monte Carlo integration with 10,000 points and the averages over trials are shown. Generally, the performance of IDA, represented by the average error rate, is significantly better (sign test, p = 0.05)

TABLE 1
The Average Error Rates (Percent) of LDA, ACC, and IDA Methods for Various Feature Space Dimensions

n	m	LDA	ACC	IDA
2	1	**47.80**	**45.88**	45.06
3	1	**48.22**	**45.02**	44.25
	2	–	**39.49**	39.04
4	1	**49.29**	**45.42**	43.90
	2	–	**38.35**	37.08
	3	–	35.07	35.08

Boldface values are significantly different from IDA error.

[4] $\sum_{i=1}^{c} P(\omega_i \mid \boldsymbol{r}) = 1$ implies $c - 1$ linearly independent features.

than those of LDA and ACC methods. Note that the advantage of IDA over LDA holds even in 1D feature space.

We will show next that these limitations of LDA and ACC can be circumvented by choosing an information-theoretic cost functional, the maximization of which gives rise to a feature extraction matrix. We will also argue that more sophisticated methods, such as the ones that can handle problems in Fig. 1, are computationally more complex than the IDA method developed here.

3 Information Discriminant Analysis

A fundamental concept in information theory is the *mutual information*. For a continuous RV $\mathrm{R} \in \mathbb{R}^n$ and a discrete class variable Ω, the mutual information, denoted by $\mu I(\mathrm{R};\Omega)$, is defined as

$$\mu I(\mathrm{R};\Omega) \triangleq H(\mathrm{R}) - H(\mathrm{R} \mid \Omega) = H(\mathrm{R}) - \sum_{i=1}^{c} H(\mathrm{R} \mid \omega_i)\, p_i, \tag{3}$$

where $H(\mathrm{R}) \triangleq -\int_{\mathcal{R}} f_{\mathrm{R}}(\boldsymbol{r}) \log(f_{\mathrm{R}}(\boldsymbol{r}))\, d\,\boldsymbol{r}$ is the differential entropy. While the mutual information primarily serves a probabilistic dependence measure, it naturally defines a multiclass feature extraction criterion [14]. Generally, the higher the mutual information, the smaller the probability of error. In particular, the following inequality[5] holds [26]:

$$\varepsilon_{\mathrm{R}} \leq \frac{1}{2}\, [H(\Omega) - \mu I(\mathrm{R};\Omega)], \tag{4}$$

where $H(\Omega)$ is the entropy of Ω. However, the practical applicability of (4) as a class-separability measure is limited by the computational cost (numerical integration in the feature space) associated with the mutual information. Next, we introduce a feature extraction objective function that is based on the mutual information, yet it is easily computable.

3.1 Information-Theoretic Objective Function

Let $f_{\mathrm{R}|\Omega}(\boldsymbol{r} \mid \omega_i)$ be a class-conditional PDF with the mean $\boldsymbol{m}_i$ and the positive definite covariance matrix $\Sigma_i > 0$, $\forall i$. To calculate the unconditional entropy $H(\mathrm{R})$, the mixture PDF, defined as $f_{\mathrm{R}}(\boldsymbol{r}) \triangleq \sum_{i=1}^{c} f_{\mathrm{R}|\Omega}(\boldsymbol{r} \mid \omega_i)\, p_i$, must be evaluated. The mean and the covariance of the mixture are given by

$$\boldsymbol{m} = \sum_{i=1}^{c} \boldsymbol{m}_i\, p_i \;\; \Sigma = \sum_{i=1}^{c} \Big[\Sigma_i + (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^{\mathrm{T}}\Big] p_i. \tag{5}$$

While the mixture PDF is not Gaussian, in general, we propose a measure similar to (3), where $H(\mathrm{R})$ is replaced by its Gaussian entropy, easily computed as $H_g(\mathrm{R}) = \frac{1}{2}\log((2\pi e)^n |\Sigma|)$. Thus, we write

$$\mu(\mathrm{R};\Omega) \triangleq H_g(\mathrm{R}) - H(\mathrm{R} \mid \Omega) = H_g(\mathrm{R}) - \sum_{i=1}^{c} H(\mathrm{R} \mid \omega_i)\, p_i. \tag{6}$$

Throughout the rest of the paper, we will refer to this criterion as the μ-measure. If we define the negentropy as $\bar{H}(\mathrm{R}) \triangleq H_g(\mathrm{R}) - H(\mathrm{R})$, the μ-measure can be decomposed as

$$\mu(\mathrm{R};\Omega) = \bar{H}(\mathrm{R}) + \mu I(\mathrm{R};\Omega). \tag{7}$$

For a fixed covariance matrix, $H_g(\mathrm{R}) \geq H(\mathrm{R})$ and the equality holds if and only if R is a Gaussian RV [27, p. 234]. Thus, the negentropy can be viewed as a measure of non-Gaussianity of a distribution. Based on (7), it follows that the μ-measure is a biased version of the mutual information, i.e., $\mu(\mathrm{R};\Omega) \geq \mu I(\mathrm{R};\Omega)$, where the bias is the negentropy. If the classes are well separated, it is clear that the resulting mutual information will be large, thereby contributing to the value of μ. Perhaps less obvious is the fact that, under these conditions, the mixture PDF is likely to be far from normal, resulting in a large $\bar{H}(\mathrm{R})$. This property of the negentropy has been used in projection pursuit applications [2], [28], where it was argued that the most interesting data projections are the ones where data appears least Gaussian. Similarly, the negentropy plays a prominent role in ICA applications, where the presence of independent signal sources may be revealed by finding these far-from-Gaussian projections. On the other hand, if the classes are fully overlapped, not only do we have $\mu I(\mathrm{R};\Omega) = 0$, but also the negentropy is likely to be relatively small.[6] In summary, the maximization of the μ-measure implies the maximization of $\bar{H}(\mathrm{R})$ and/or $\mu I(\mathrm{R};\Omega)$, both of which are proper class-separability criteria. We will refer to the maximization of the μ-measure over a subspace of a fixed dimension (see Section 3.6) as information discriminant analysis.

3.2 Theoretical Properties of the μ-Measure

The μ-measure shares many properties of the Bayes error ε_{R} and the mutual information (3).

Theorem 1 (Subspace Invariance). *The μ-measure is invariant under invertible linear transformations, i.e., $\mu(\bar{\mathrm{R}};\Omega) = \mu(\mathrm{R};\Omega)$, where $\bar{\mathrm{R}} \triangleq \boldsymbol{T}\,\mathrm{R}$, and $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^n$ is a nonsingular transformation matrix.*

Proof. For any nonsingular matrix, $\boldsymbol{T}$, we have $H(\boldsymbol{T}\,\mathrm{R}) = H(\mathrm{R}) + \log(|\boldsymbol{T}|)$, where $|\,.\,|$ is the absolute value of the determinant [27, p. 234]. It follows from the definition (6) that

$$\mu(\bar{\mathrm{R}};\Omega) = H_g(\mathrm{R}) + \log(|\boldsymbol{T}|) - [H(\mathrm{R} \mid \Omega) + \log(|\boldsymbol{T}|)] = \mu(\mathrm{R};\Omega).$$

□

Theorem 2. *For any full-rank transformation matrix $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^m$ $(m < n)$, there exists an orthonormal[7] matrix $\boldsymbol{E} \in \mathbb{R}^{m\times n}$, so that, if $\bar{\mathrm{R}} \triangleq \boldsymbol{T}\,\mathrm{R}$ and $\tilde{\mathrm{R}} \triangleq \boldsymbol{E}\,\mathrm{R}$, we have $\mu(\bar{\mathrm{R}};\Omega) = \mu(\tilde{\mathrm{R}};\Omega)$.*

Proof. By the Singular Value Decomposition Theorem, there exist orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ such that $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{T}\,\boldsymbol{V} = [\Lambda \mid \mathbf{0}_{m\times d}]$, where $\Lambda \in \mathbf{R}^{m\times m}$ is a diagonal matrix of singular values of $\boldsymbol{T}$ and $d \triangleq n - m$. Note that $\mathrm{rank}(\boldsymbol{T}) = m$ implies that Λ is invertible. Define $\bar{\mathrm{R}} \triangleq \boldsymbol{T}\,\mathrm{R}$ and $\tilde{\mathrm{R}} \triangleq \Lambda^{-1}\boldsymbol{U}^{\mathrm{T}}\bar{\mathrm{R}}$ and note that $\mu(\bar{\mathrm{R}};\Omega) = \mu(\tilde{\mathrm{R}};\Omega)$ based on Theorem 1. From the definitions of $\bar{\mathrm{R}}$ and $\tilde{\mathrm{R}}$, it follows that $\tilde{\mathrm{R}} = \Lambda^{-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{T}\,\mathrm{R}$. Define $\boldsymbol{E} \triangleq \Lambda^{-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{T} \in \mathbb{R}^{m\times n}$ and note that $\boldsymbol{E}\,\boldsymbol{V} = [\boldsymbol{I}_m \mid \mathbf{0}_{m\times d}]$. It follows readily that $\boldsymbol{E}\,\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{E}^{\mathrm{T}} = \boldsymbol{I}_m$, which, after recalling that $\boldsymbol{V}$ is an orthogonal matrix, completes the proof. □

[5] Hellman and Raviv [26] give examples of distributions where (4) does not hold. These pathological cases will not be treated here.

[6] Examples may be constructed where nonoverlapping classes give rise to a Gaussian mixture and fully overlapped classes give rise to a far-from-Gaussian mixture, but these are of theoretical significance and rarely appear in practical applications.

[7] Matrix $\boldsymbol{E} \in \mathbb{R}^{m\times n}$ is orthonormal if $\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}} = \boldsymbol{I}_m$, where $\boldsymbol{I}_m$ is the identity matrix of size m.

Theorem 3 (Data Processing Inequality). *The μ-measure is nonincreasing under linear transformations, i.e.,* $\mu(\bar{\mathrm{R}};\Omega) \leq \mu(\mathrm{R};\Omega)$, *where* $\bar{\mathrm{R}} \triangleq \boldsymbol{T}\,\mathrm{R}$ *and* $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^m$ $(m < n)$ *is a full-rank transformation matrix.*

Proof. Let $\boldsymbol{T}^{\perp} \in \mathbb{R}^{(n-m)\times n}$ be a matrix whose rows span the null space of $\boldsymbol{T}$. Define the matrix $\tilde{\boldsymbol{T}}$ and the RV $\tilde{\mathrm{R}}$ as

$$\tilde{\boldsymbol{T}} \triangleq \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{T}^{\perp} \end{bmatrix} \in \mathbb{R}^{n\times n},\ \tilde{\mathrm{R}} \triangleq \tilde{\boldsymbol{T}}\,\mathrm{R} = \begin{bmatrix} \boldsymbol{T}\,\mathrm{R} \\ \boldsymbol{T}^{\perp}\mathrm{R} \end{bmatrix} \triangleq \begin{bmatrix} \bar{\mathrm{R}} \\ \bar{\mathrm{N}} \end{bmatrix}.$$

Since $\tilde{\boldsymbol{T}}$ is a full-rank matrix, it follows from Theorem 1 that $\mu(\tilde{\mathrm{R}};\Omega) = \mu(\mathrm{R};\Omega)$. From the definition of $\tilde{\mathrm{R}}$, we have

$$\mu(\tilde{\mathrm{R}};\Omega) = \mu(\bar{\mathrm{R}},\bar{\mathrm{N}};\Omega) = H_g(\bar{\mathrm{R}},\bar{\mathrm{N}}) - H(\bar{\mathrm{R}},\bar{\mathrm{N}} \mid \Omega).$$

We write based on the chain rule for entropies and conditional entropies [27, p. 232]

$$\begin{aligned}\mu(\tilde{\mathrm{R}};\Omega) &= H_g(\bar{\mathrm{R}}) + H_g(\bar{\mathrm{N}} \mid \bar{\mathrm{R}}) - [H(\bar{\mathrm{R}} \mid \Omega) + H(\bar{\mathrm{N}} \mid \bar{\mathrm{R}},\Omega)] \\ &= \mu(\bar{\mathrm{R}};\Omega) + H_g(\bar{\mathrm{N}} \mid \bar{\mathrm{R}}) - H(\bar{\mathrm{N}} \mid \bar{\mathrm{R}},\Omega) \geq \mu(\bar{\mathrm{R}};\Omega),\end{aligned}$$

where the inequality follows by noting that $H_g(\bar{\mathrm{N}} \mid \bar{\mathrm{R}}) \geq H(\bar{\mathrm{N}} \mid \bar{\mathrm{R}}) \geq H(\bar{\mathrm{N}} \mid \bar{\mathrm{R}},\Omega)$ (Gaussian distribution maximizes entropy and conditioning reduces entropy [27, p. 232]). □

We comment briefly on the importance of the above theorems. Theorem 1 simply states that, like many other discriminant measures, such as the Bayes error or the mutual information, the μ-measure is independent of the choice of a coordinate system for data representation. The implication of Theorem 2 is that the search for a full-rank feature extraction matrix $\boldsymbol{T}$ can be restricted to a subspace of orthonormal projection matrices without compromising the objective function. While theoretical performances of orthonormal and arbitrary projection matrices are equivalent, in practical applications, these oblique projections may cause awkward scalings of data and may lead to numerical instability. Finally, Theorem 3 states that the μ-measure of any subspace of the original data space is bounded above by the μ-measure of the original space.

3.3 The Calculation of the μ-Measure

To complete the calculation of μ given by (6), the class-conditional PDFs and, in turn, the corresponding entropies, $H(\mathrm{R} \mid \omega_i)$, must be evaluated. In principle, this can be achieved using either a parametric (model-based) or a nonparametric (kernel-based) approach. The advantage of the parametric approach is that computations are simpler, although the assumed model may not be supported by data. The nonparametric approach, on the other hand, does not assume any particular model, but analytical tractability is inevitably lost, and computations are much more demanding. Interestingly, Devijver and Kittler [14] argue against the use of nonparametric density estimates on the grounds of both simplicity and accuracy, claiming that, in the face of a limited sample size, the errors on the nonparametric PDF estimate may by far exceed those of simple parametric models, such as Gaussians. Throughout the rest of the paper, we will use the parametric estimates of the class-conditional entropies $H(\mathrm{R} \mid \omega_i)$. In particular, the entropies are modeled as

$$H(\mathrm{R} \mid \omega_i) = \frac{1}{2}\log((2\pi e)^n|\Sigma_i|),$$

which coupled with (6) yields a very simple version of the μ-measure

$$\mu(\mathrm{R},\Omega) = \frac{1}{2}\left[\log(|\Sigma|) - \sum_{i=1}^{c}\log(|\Sigma_i|)\,p_i\right]. \qquad (8)$$

While (8)[8] is ideally suited for Gaussian classes, we will argue below, and demonstrate by numerous examples, that it is also a general class-separability criterion. Furthermore, because of its analytical tractability, we will show how μ relates to known statistical quantities and how it can be efficiently maximized. Like IDA and ACC, the criterion (8) employs the first two statistical moments and belongs to the category of second-order criteria.

First note that the primary goal of the criterion (8) is to measure the class separability, rather than to approximate the mutual information. If the class differences are captured by the first two statistical moments, it is expected that the μ-measure (8) will perform well. Fig. 1 demonstrates that this is indeed the case; even if the class-conditional PDFs are multimodal and far from Gaussian (Figs. 1b and 1c), the criterion (8) performs optimally. In addition, the μ-measure may be optimal even if the support of the PDF is not simply connected (Fig. 1c), and even if the classes have discrete distributions (Fig. 1d). On the other hand, LDA and ACC fail completely, as illustrated by the Bayes error of 50 percent.

While classes with fully overlapped means are unlikely to be found in real situations, a simple continuity argument suggests that IDA will retain the advantage over LDA and ACC if the class-conditional means remain relatively close. To test this hypothesis, we performed 1,000 Monte Carlo trials for examples (Figs. 1a, 1b, and 1c) and calculated the Bayes error using a numerical integration. Briefly, the class covariances were randomized and the first class was centered at the origin. The second class was rotated by a random angle, sampled uniformly from $[0, 2\pi]$, and the class was centered ρ Mahalanobis distances from the origin. Therefore, we have $\boldsymbol{m}_2{}^{\mathrm{T}}\,\Sigma_1{}^{-1}\boldsymbol{m}_2 = \rho^2$, where ρ was sampled uniformly from $[0, D]$. All random samplings were performed on a trial-by-trial basis. Table 2 shows the average error rates and the average values of μ for several values of D and for the following feature extraction techniques: LDA, ACC, IDA, and the Patrick-Fisher (PF) measure, which was chosen as a computationally feasible representative of probabilistic dependence measures [14, p. 261]. The performance of IDA is uniformly superior (the smallest ε) to those of LDA, ACC, and PF methods. For small values of D, the performances of PF and IDA are similar, but they seem to diverge as D increases, with the performance margin increasing in favor of IDA. On the other hand, IDA holds a substantial advantage over LDA and ACC when the means are relatively close and the margin of improvement decreases with D. As classes are more separated, the three techniques yield errors which are more similar. Note that μ is negatively correlated with ε; hence, its use as a class-separability measure is justified. In most of the cases, the advantage of IDA over the other three techniques is statistically significant (sign test, p = 0.05).

In summary, if the class-conditional means are relatively close, the μ-measure may provide significant improvement

[8] It was brought to the author's attention at the time of publication of this manuscript that (8) was used earlier in [47].

TABLE 2
The Average Error Rates (Percent) and the Average Values of the μ-Measure for Various Examples (Fig. 1)

Example		(a)				(b)				(c)			
D	Method	LDA	ACC	IDA	PF	LDA	ACC	IDA	PF	LDA	ACC	IDA	PF
0.5	ε	**39.15**	**39.47**	35.76	36.13	**31.70**	**31.82**	27.19	27.92	**32.47**	**33.10**	25.45	25.47
	μ	**0.077**	**0.068**	0.122	0.121	**0.127**	**0.116**	0.181	**0.167**	**0.108**	**0.099**	0.179	0.175
1.0	ε	**35.80**	**36.11**	33.06	**35.16**	**29.94**	**30.28**	26.20	26.99	**31.11**	**30.91**	24.83	25.07
	μ	**0.109**	**0.101**	0.153	**0.127**	**0.163**	**0.152**	0.215	**0.196**	**0.142**	**0.137**	0.216	0.196
2.0	ε	**29.13**	**29.76**	27.20	**30.79**	**24.83**	**25.40**	22.57	**24.94**	**27.44**	**27.76**	23.52	**24.83**
	μ	**0.194**	**0.179**	0.233	**0.180**	**0.272**	**0.253**	0.319	**0.258**	**0.230**	**0.218**	0.297	**0.240**
3.0	ε	**22.73**	**23.29**	21.30	**26.31**	**19.72**	**20.15**	18.08	**21.62**	**22.77**	**22.92**	19.79	**22.14**
	μ	**0.310**	**0.295**	0.348	**0.260**	**0.383**	**0.365**	0.426	**0.330**	**0.339**	**0.325**	0.400	**0.311**

Boldface values are significantly different from IDA values.

over other second-order techniques such as LDA and ACC. We will see next that this property of the μ-measure can be explained theoretically.

3.4 The μ-Measure as a Measure of Class Separability

Ideally, the μ-measure should be linked to the Bayes error in a manner similar to (4). A useful connection between $\mu(\mathrm{R}, \Omega)$ and ε_{R}, however, is not easy to establish even if the simplified definition (8) is used. We will argue next that the μ-measure increases with class separability. Before we proceed, note that, under the homoscedastic conditions $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_c$, the definition (8) reduces to

$$\mu(\mathrm{R}, \Omega) = \frac{1}{2}\log\frac{|\Sigma_{\mathrm{B}} + \Sigma_{\mathrm{W}}|}{|\Sigma_{\mathrm{W}}|},$$

which is a variant of LDA [1, p. 446], where $\Sigma_{\mathrm{B}} \triangleq \sum_{i=1}^{c}[(\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^{\mathrm{T}}]\, p_i$ and $\Sigma_{\mathrm{W}} \triangleq \sum_{i=1}^{c} \Sigma_i\, p_i$ are the between and within-class scatter matrices, respectively. Therefore, under the homoscedastic conditions, IDA reduces to the classical LDA. In addition, if the classes are Gaussian, the two methods will be optimal the sense of Bayes (see Section 2).

Another interesting point is that, for a binary Gaussian class case with a relatively small difference in means, the μ-measure reduces to yet another well-known statistical measure—the Chernoff distance. To see this, rewrite (8) as

$$\mu(\mathrm{R}; \Omega) = \frac{1}{2}\log\frac{|\Sigma_{\mathrm{W}} + (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^{\mathrm{T}}\; p_1 p_2|}{\prod_{i=1}^{2} |\Sigma_i|^{p_i}},$$

which follows readily after noting that $\boldsymbol{m} = \sum_{i=1}^{2} \boldsymbol{m}_i\, p_i$ and $\Sigma_{\mathrm{W}} = \sum_{i=1}^{2} \Sigma_i\, p_i$. Since the second term in the numerator is a rank-1 matrix, the expression above can be simplified as

$$\mu(\mathrm{R}; \Omega) = \frac{1}{2}\log\frac{|\Sigma_{\mathrm{W}}|}{\prod_{i=1}^{2} |\Sigma_i|^{p_i}} + \frac{1}{2}\log\Big(1 + (\boldsymbol{m}_1 - \boldsymbol{m}_2)^{\mathrm{T}}\Sigma_{\mathrm{W}}^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2) p_1 p_2\Big).$$

For $\|\boldsymbol{x}\| \approx 0$, the following second order Taylor series expansion is valid

$$\log(1 + \alpha\, \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}) \approx \alpha\, \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$$

and, so, we finally write

$$\mu(\mathrm{R}; \Omega) = \frac{p_1 p_2}{2}(\boldsymbol{m}_1 - \boldsymbol{m}_2)^{\mathrm{T}}\Sigma_{\mathrm{W}}^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2) + \frac{1}{2}\log\frac{|\Sigma_{\mathrm{W}}|}{\prod_{i=1}^{2} |\Sigma_i|^{p_i}}, \tag{9}$$

which is the Chernoff distance, $\rho(s)$, evaluated at $s = p_1$ [1, p. 99]. Therefore, for two heteroscedastic Gaussian classes with relatively close means, the μ-measure approaches the Chernoff distance, thereby defining the upper bound for the Bayes error, i.e., $\varepsilon_{\mathrm{R}} \leq p_1^{p_1} p_2^{p_2} e^{-\mu(\mathrm{R};\Omega)}$. This explains the superior performance of IDA on the example from Fig. 1a and the related Monte Carlo simulations, especially at low values of D (see Table 2). For non-Gaussian classes, the expression (9) no longer represents the Chernoff distance, although this criterion has been successfully used in many pattern recognition problems [16], [15] or as a basis for the development of novel feature extracting methods [17]. Therefore, the good performance of IDA in examples Figs. 1b and 1c is not surprising even though the classes are far from Gaussian.

Let us now show that the μ-measure is a valid class-separability measure under more general conditions. Suppose first that $c = 2$, $\boldsymbol{m}_1 = \boldsymbol{m}_2$, and $\Sigma_1 = \Sigma_2$. It follows from (5) and (8) that $\mu = 0$. Adding a little perturbation to $\boldsymbol{m}_1$ and/or Σ_1 would result in a μ that is slightly positive. Our conjecture is that by "increasing" these perturbations, the μ-measure would also increase, therefore capturing the differences between $\{\boldsymbol{m}_1, \Sigma_1\}$ and $\{\boldsymbol{m}_2, \Sigma_2\}$. For the sake of simplicity, we consider two separate cases.

3.4.1 *Fixed Covariances*

Suppose $\Sigma_1,\ \Sigma_2, \cdots,\ \Sigma_c$ are fixed and define deviations $\tilde{\boldsymbol{m}}_i \triangleq \boldsymbol{m}_i - \boldsymbol{m}, \forall i$. In the interest of clarity, we will study the variation of a single vector, say $\boldsymbol{m}_j$. The analysis extends to multiple vectors in a straightforward fashion. Therefore, assume that all $\boldsymbol{m}_i$ are fixed with the exception of $\boldsymbol{m}_j$. Also, assume that $\boldsymbol{m}_j \neq \boldsymbol{m}$, i.e., $\tilde{\boldsymbol{m}}_j \neq \boldsymbol{0}$, where $\boldsymbol{0}$ stands for a zero vector in $\mathbb{R}^n$. We want to argue that moving in the direction of $\tilde{\boldsymbol{m}}_j$ can only increase the function μ. An illustrative example with Gaussian classes is shown in Fig. 2.

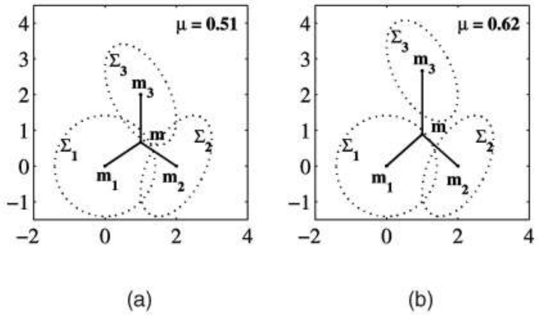

Fig. 2. (a) Three Gaussian classes. (b) Moving mean $\boldsymbol{m}_3$ in the direction of $\tilde{\boldsymbol{m}}_3$ reduces the overlap, thus increasing the class separability and the μ-measure.

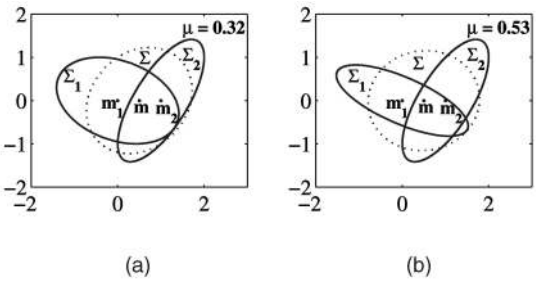

Fig. 3. (a) Two Gaussian classes. (b) Changing the covariance Σ_1 in the "direction" of $\tilde{\Sigma}_1$ reduces the overlap and increases the class separability and μ.

Proposition 1.

$$\left\langle \frac{\partial \mu}{\partial \tilde{\boldsymbol{m}}_j}, \tilde{\boldsymbol{m}}_j \right\rangle > 0, \tag{10}$$

where $\partial\mu/\partial\tilde{\boldsymbol{m}}_j \in \mathbb{R}^n$ is the gradient of μ with respect to $\tilde{\boldsymbol{m}}_j$ and $\langle\, , \rangle$ is an inner product in $\mathbb{R}^n$. The proof of this proposition is given in the supplemental Appendix A, which can be found at http://computer.org/tpami/archives.htm.

3.4.2 Fixed Means

Suppose now that $\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_c$ are fixed. Define matrix deviations $\tilde{\Sigma}_i \overset{\triangle}{=} \Sigma_i - \Sigma$, and assume that all Σ_i are fixed with the exception of a single matrix, say Σ_j. A similar argument extends to multiple matrices. Also, assume that $\Sigma_j \neq \Sigma$, i.e., $\tilde{\Sigma}_j \neq \mathbf{0}_n$, where $\mathbf{0}_n$ stands for a zero matrix in $\mathbb{R}^{n\times n}$. We want to show that moving in the direction of $\tilde{\Sigma}_j$ can only increase the function μ. An example of this situation is illustrated by Fig. 3.

Proposition 2.

$$\langle \partial\mu/\partial\tilde{\Sigma}_j : \tilde{\Sigma}_j \rangle > 0, \tag{11}$$

where $\partial\mu/\partial\tilde{\Sigma}_j \in \mathbb{R}^{n\times n}$ is the gradient of μ with respect to $\tilde{\Sigma}_j$ and $\langle : \rangle$ stands for an inner product of matrices, defined as $\langle \boldsymbol{A} : \boldsymbol{B} \rangle \overset{\triangle}{=} \sum_{i,j} A_{i,j}\, B_{i,j}$ for two arbitrary matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same size. The proof of this proposition is given in the supplemental Appendix B, which can be found at http://computer.org/tpami/archives.htm.

We finish this section by commenting briefly on the two propositions. Since the projection of the gradient of μ to the parameter vector is strictly positive, moving in the direction of the parameter vector can only increase the μ-measure. By definition, the parameter vector is proportional to the class separability, hence, we conclude that the μ-measure is a valid class-separability measure.

3.5 Bayes Optimality of IDA

We have seen in Section 3.4 that, when Gaussian classes conform to the homoscedastic conditions, IDA is an optimal feature extraction technique in the sense of Bayes. We will now give more general (heteroscedastic) conditions under which IDA is optimal. Let $\mathrm{R} \in \mathbb{R}^n$ be a data vector. Let us assume that the class differences are confined to an mD subspace in $\mathbb{R}^n$, and that classes are fully overlapped in the complementary dD space ($d \overset{\triangle}{=} n - m$). We will refer to these subspaces as the signal subspace and the noise subspace, respectively. Assume further that class data is a linear mixture of a Gaussian signal $\mathrm{S}|\Omega \in \mathbb{R}^m$ and a Gaussian noise $\mathrm{N}|\Omega \in \mathbb{R}^d$

$$\mathrm{R} \mid \omega_i = \boldsymbol{M} \begin{bmatrix} \mathrm{S} \mid \omega_i \\ \mathrm{N} \mid \omega_i \end{bmatrix} \quad \forall i = \{1, 2, \cdots, c\}, \tag{12}$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is a nonsingular mixing matrix. This type of model is frequently used in *array signal processing* [29], where the number of sensors is larger than the number of signal sources.

Theorem 4. *Let $\boldsymbol{T} \in \mathbb{R}^{m\times n}$ be a full-rank matrix that maximizes $\mu(\bar{\mathrm{R}}; \Omega)$, where $\bar{\mathrm{R}} \overset{\triangle}{=} \boldsymbol{T}\,\mathrm{R}$. If $f_{\mathrm{N}|\mathrm{S},\Omega}(\boldsymbol{n} \mid \boldsymbol{s}, \omega_i) = f_{\mathrm{N}|\mathrm{S}}(\boldsymbol{n} \mid \boldsymbol{s})$, $\forall i = \{1, 2, \cdots, c\}$, $\forall \boldsymbol{s} \in \mathbb{R}^m$, $\forall \boldsymbol{n} \in \mathbb{R}^d$, where $\mathrm{S}|\Omega$ and $\mathrm{N}|\Omega$ are Gaussian random variables, then $\boldsymbol{T}$ extracts the signal subspace, i.e., $\mu(\bar{\mathrm{R}}; \Omega) = \mu(\mathrm{S}; \Omega)$. Moreover, such a subspace is optimal in the sense of Bayes, i.e., $\varepsilon_{\bar{\mathrm{R}}} = \varepsilon_{\mathrm{R}}$.*

Proof. Without loss of generality, assume $\boldsymbol{M} = \boldsymbol{I}_n$. If not, we can always find a transformation matrix $\tilde{\boldsymbol{T}} = \boldsymbol{M}^{-1}$ such that

$$\tilde{\mathrm{R}} \overset{\triangle}{=} \tilde{\boldsymbol{T}}\,\mathrm{R} = \begin{bmatrix} \mathrm{S} \\ \mathrm{N} \end{bmatrix},$$

where $\mu(\tilde{\mathrm{R}}; \Omega) = \mu(\mathrm{R}; \Omega)$ (by Theorem 1) and $\varepsilon_{\tilde{\mathrm{R}}} = \varepsilon_{\mathrm{R}}$ (see Section 2). Therefore, we have

$$\mu(\mathrm{R}; \Omega) = \mu(\mathrm{S}, \mathrm{N}; \Omega) = \mu(\mathrm{S}; \Omega) + H_g(\mathrm{N} \mid \mathrm{S}) - H(\mathrm{N} \mid \mathrm{S}, \Omega), \tag{13}$$

which follows after applying the chain rule for entropies. From the conditions of the theorem, it follows that N is independent of Ω, given S, thus $H(\mathrm{N} \mid \mathrm{S}, \Omega) = H(\mathrm{N} \mid \mathrm{S})$. To prove that $\mu(\mathrm{R}; \Omega) = \mu(\mathrm{S}; \Omega)$ in (13), it suffices to show that N|S is Gaussian. Since both S|Ω and N|Ω are Gaussian by assumption, so is N|S, Ω. From $f_{\mathrm{N}|\mathrm{S}}(\boldsymbol{n} \mid \boldsymbol{s}) = f_{\mathrm{N}|\mathrm{S},\Omega}(\boldsymbol{n} \mid \boldsymbol{s}, \omega_i)$, it follows that N|S is Gaussian. Thus, we have $\mu(\mathrm{R}; \Omega) = \mu(\mathrm{S}; \Omega)$, which, combined with Theorem 3 and the assumptions of Theorem 4, yields

$$\mu(\mathrm{R}; \Omega) \geq \mu(\bar{\mathrm{R}}; \Omega) \geq \mu(\mathrm{S}; \Omega) = \mu(\mathrm{R}; \Omega).$$

Next, we show that $\varepsilon_{\mathrm{S}} = \varepsilon_{\mathrm{R}}$ by showing that the Bayes assignment is preserved, i.e.,

$$\begin{aligned} P(\omega_i \mid \boldsymbol{r}) &= \frac{f_{\mathrm{R}|\Omega}(\boldsymbol{r} \mid \omega_i)\; p_i}{f_{\mathrm{R}}(\boldsymbol{r})} = \frac{f_{\mathrm{N}|\mathrm{S},\Omega}(\boldsymbol{n} \mid \boldsymbol{s}, \omega_i) f_{\mathrm{S}|\Omega}(\boldsymbol{s} \mid \omega_i)\; p_i}{f_{\mathrm{R}}(\boldsymbol{r})} \\ &= \frac{f_{\mathrm{N}|\mathrm{S}}(\boldsymbol{n} \mid \boldsymbol{s}) f_{\mathrm{S}|\Omega}(\boldsymbol{s} \mid \omega_i)\; p_i}{f_{\mathrm{N},\mathrm{S}}(\boldsymbol{n}, \boldsymbol{s})} = \frac{f_{\mathrm{S}|\Omega}(\boldsymbol{s} \mid \omega_i)\; p_i}{f_{\mathrm{S}}(\boldsymbol{s})} \\ &= P(\omega_i \mid \boldsymbol{s}) \quad \forall i = \{1, 2, \cdots, c\}, \forall \boldsymbol{r} \in \mathbb{R}^n. \end{aligned}$$

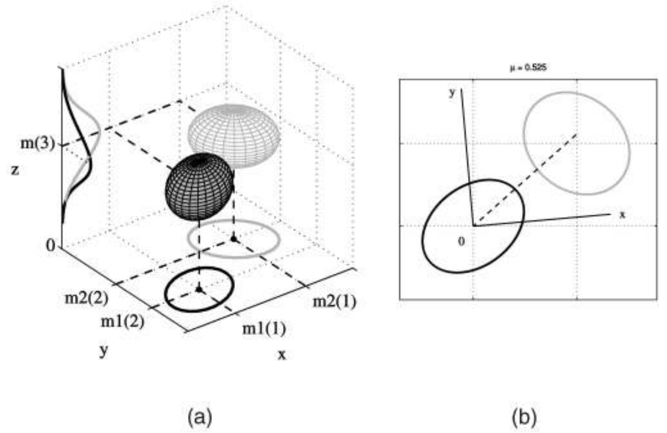

Fig. 4. (a) Two Gaussian classes in 3D with the corresponding orthogonal projections (ellipses). (b) The IDA-optimal 2D subspace together with $x0y$ plane. The dashed line marks the optimal 1D subspace extracted by LDA.

Since the Bayes error is subspace invariant (see Section 2), it follows that $\varepsilon_{\bar{\mathrm{R}}} = \varepsilon_{\mathrm{S}}$. □

Before we proceed, let us comment briefly on this result. First, Theorem 4 gives sufficient conditions for the optimality of IDA. Second, apart from the fact that the class Ω has no bearing on the noise N when the signal S is given, these conditions are somewhat hard to interpret. Fortunately, since Gaussian RVs are involved, it is easy to find another set of conditions which are sufficient for optimality of IDA, yet they are easy to interpret. The following corollary of Theorem 4 states that, if the class differences are confined to the signal subspace and if the noise and signal are uncorrelated over classes, then IDA is an optimal DFE method in the sense of Bayes, and will extract the signal subspace. Kumar and Andreou [10] proposed an ML solution of the problem above, although they did not explicitly show that their solution extracts the signal subspace.

Corollary 1. *Let $\mathrm{R}|\Omega$ be a Gaussian RV defined as in (12) with $\boldsymbol{M} = \boldsymbol{I}_n$ and with*

$$\boldsymbol{m}_i = \begin{bmatrix} \boldsymbol{m}_i^{\mathrm{S}} \\ \boldsymbol{m}_i^{\mathrm{N}} \end{bmatrix} \quad \Sigma_i = \begin{bmatrix} \Sigma_i^{\mathrm{SS}} & \Sigma_i^{\mathrm{SN}} \\ \Sigma_i^{\mathrm{NS}} & \Sigma_i^{\mathrm{NN}} \end{bmatrix}.$$

Let $\boldsymbol{T} \in \mathbb{R}^{m \times n}$ be a full-rank matrix that maximizes $\mu(\bar{\mathrm{R}};\Omega)$, where $\bar{\mathrm{R}} \overset{\triangle}{=} \boldsymbol{T}\,\mathrm{R}$. If $\boldsymbol{m}_i^{\mathrm{N}} = \boldsymbol{m}^{\mathrm{N}}$, $\Sigma_i^{\mathrm{NN}} = \Sigma^{\mathrm{NN}}$, and $\Sigma_i^{\mathrm{SN}} = (\Sigma_i^{\mathrm{NS}})^{\mathrm{T}} = \mathbf{0}_{m \times d}, \forall i$, then this transformation extracts the signal subspace, i.e., $\mu(\bar{\mathrm{R}};\Omega) = \mu(\mathrm{S};\Omega)$. Moreover, such a subspace is optimal in the sense of Bayes.

The proof of this corollary is given in the supplemental Appendix E, which can be found at http://computer.org/tpami/archives.htm and an illustrative example is shown in Fig. 4.

The signal subspace is the $x0y$-plane and the noise subspace is the z-axis (note the overlap in PDFs along this axis). Fig. 4 also shows that IDA estimates the optimal 2D subspace up to a rotation matrix. Finally, note that, if Corollary 1 holds for RV R, it also holds for RV $\tilde{\mathrm{R}} \overset{\triangle}{=} \boldsymbol{M}\,\mathrm{R}$, where $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ is an arbitrary nonsingular matrix.

3.6 Maximization of the μ-Measure

From the three theorems in Section 3.2, it follows that the following optimization problem is well-posed. Given the data $\mathrm{R} \in \mathbb{R}^n$ and the dimension, m, of a feature space, we wish to find a full-rank matrix $\boldsymbol{T} \in \mathbb{R}^{m \times n}$ such that the measure $\mu(\bar{\mathrm{R}};\Omega)$ is maximized, i.e.,

$$\begin{aligned} \boldsymbol{T}^* = \arg \max_{\boldsymbol{T} \in \mathbb{R}^{m \times n}} &\{\mu(\bar{\mathrm{R}};\Omega) \ : \ \bar{\mathrm{R}} = \boldsymbol{T}\,\mathrm{R}\} \\ &\text{subject to } \boldsymbol{T}\boldsymbol{T}^{\mathrm{T}} = \boldsymbol{I}_m, \end{aligned} \tag{14}$$

where, based on Theorem 2, we have replaced the condition $\mathrm{rank}(\boldsymbol{T}) = m$ in (14) with the constraint $\boldsymbol{T}\,\boldsymbol{T}^{\mathrm{T}} = \boldsymbol{I}_m$. From (8) and (22) (see the supplemental Appendix C, which can be found at http://computer.org/tpami/archives.htm), the gradient $\partial\mu/\partial\boldsymbol{T} \in \mathbb{R}^{m \times n}$ can be found as

$$\frac{\partial\mu(\bar{\mathrm{R}};\Omega)}{\partial\boldsymbol{T}} = (\boldsymbol{T}\,\Sigma\boldsymbol{T}^{\mathrm{T}})^{-1}\boldsymbol{T}\,\Sigma - \sum_{i=1}^{c}(\boldsymbol{T}\,\Sigma_i\boldsymbol{T}^{\mathrm{T}})^{-1}\boldsymbol{T}\,\Sigma_i\,p_i. \tag{15}$$

Unfortunately, the equation $\partial\mu(\bar{\mathrm{R}};\Omega)/\partial\boldsymbol{T} = 0$ cannot be solved analytically, so the maximization (14) must be performed numerically using gradient-based optimization schemes. However, the Hessian $\partial^2\mu/\partial\boldsymbol{T}^2 \in \mathbb{R}^{mn \times mn}$ can be found analytically as

$$\frac{\partial^2\mu(\bar{\mathrm{R}};\Omega)}{\partial\boldsymbol{T}^2} = \boldsymbol{A}(\Sigma) + \boldsymbol{B}(\Sigma) - \sum_{i=1}^{c}[\boldsymbol{A}(\Sigma_i) + \boldsymbol{B}(\Sigma_i)]p_i, \tag{16}$$

where the matrix functions $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined by (23) (see Appendix C). Therefore, the maximization (14) is amenable to Newton-based optimization routines. While posed as a constrained optimization problem, (14) can be solved using an unconstrained approach, where, at each iteration, the current solution is projected to the constraint set (see Theorem 2).

4 Experimental Results

The performance of IDA was tested experimentally and compared to those of the LDA, ACC, and PF methods. Both LDA and ACC yield feature extraction matrices analytically, therefore, they are easy to implement. For the PF criterion, a parametric model of class-conditional PDFs was used, so that the four criteria are comparable in that they are all second-order techniques. The IDA and PF extraction matrices were found numerically.

The performances were tested against five data sets taken from the UCI machine learning repository [30] and a data set adopted from [31]. These data sets come from a variety of applications, with various numbers of classes and attributes and various sample sizes (see Table 3). The instances with missing values in data set (b) were ignored throughout the experiments. The performances were evaluated based on the linear and quadratic Bayesian classifiers [25, p. 39], with prior probabilities estimated empirically from the data. In addition, for data sets (e) and (f), a support vector machine (SVM) implementation, SVMTorch [32], was used, and all experiments were performed with a Gaussian kernel. For these two data sets, training and test data were specifically designated by their donors; therefore, the DFE matrices and the classifier parameters were estimated from the training data and validated on the test data. The number of test instances is given in Table 3. For all other data sets, a k-fold cross-validation (CV) was used. The justification of the choice of k will be given below.

TABLE 3
Data Sets Used for Performance Assessment

Data set	Label	n	c	N_i	Validation
Brain	(a)	8	2	162	10-fold
Heart	(b)	13	2	296	10-fold
Balance	(c)	4	3	625	20-fold
Vehicle	(d)	18	4	846	20-fold
Satellite	(e)	36	6	6435	2000
Letter	(f)	16	26	20000	4000

The columns are: n is the number of attributes, c is the number of classes, N_i is the number of instances, and Validation is the type of validation or the number of test data.

4.1 Performance Evaluation

For data sets (e) and (f), the transformation matrices $\boldsymbol{T} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ were estimated from the training data, which was then transformed by $\boldsymbol{T}$ to a subspace of appropriate dimension, where the two classifiers were designed. The test data was then transformed by $\boldsymbol{T}$ and the probability of misclassification (error) was computed. Note that the admissible subspace dimensions are $1 \leq m \leq c - 1$ for LDA and $1 \leq m \leq n - 1$ for the IDA and ACC methods. For data sets (a), (b), (c), and (d), the evaluation consists of the following procedure:

1. Randomly divide the data set into k nonoverlapping folds of equal size.
2. Designate one of the folds as test data and combine the other $k - 1$ folds (and possible leftover data) into a single training set.
3. Based on the training set, compute the transformation matrix to a subspace of appropriate dimension, m, and design the classifiers based on the transformed training data. Transform the test data in the same manner, perform the classification, and log the number of misclassified instances.
4. Repeat Steps 2 and 3 until all the folds are exhausted. Estimate the error as the total number of misclassified cases divided by the total number of test instances [33].
5. Repeat Steps 1 through 4 several (5-10) times to obtain multiple estimates of the error.

The sample mean of the error obtained above is used as the estimate of the probability of misclassification, and its sample variance is used to construct the confidence intervals.

The choice of the number of folds, k, for CV is dictated by the bias-variance trade-off. For small values of k, a large discrepancy in size between the training set and the data set may overestimate the probability of error [33]. On the other hand, for $k = N_i$, also known as leave-one-out CV, the error estimates are almost unbiased, but often with unacceptably high variability [34]. Also, for data with a large number of instances, N_i, leave-one-out CV can be computationally demanding. It is widely accepted that 10 to 20-fold CV offers a good bias-variance compromise, and these values are often used as default. As suggested in [33], further improvements in terms of both bias and variance can be obtained by stratification, where the relative class frequencies over folds roughly match those of the original data set. Therefore, for large data sets ($N_i > 500$), a stratified 20-fold CV was used (see Table 3). Other data sets were tested using a stratified 10-fold CV.

4.2 Analysis of Results

We begin with the analysis of data sets (e) and (f), where only a single cross-validatory run was performed. The respective results are given in Table 4 and Table 5. The errors are calculated based on the four DFE methods and the three classifiers: linear (L), quadratic (Q), and SVM (S). The performance of the classifiers in the original space (FULL) is also shown. In the interest of space, the results for selected subspaces are shown, featuring the smallest error for each method-classifier combination. These optimal error values are shown in bold. The best performance for a given classifier

TABLE 4
The Estimated Error (Percent) for Various DFE Method-Classifier Combinations for Data Set Satellite (e)

Method	Classifier	Size of feature space (m)							
		4	5	19	20	27	31	33	36 (FULL)
LDA	L	17.25	**17.15**	–	–	–	–	–	**17.15**
	Q	**15.30**	15.50	–	–	–	–	–	15.20
	S (28)	12.30	**12.20**	–	–	–	–	–	8.05
ACC	L	17.80	17.75	17.00	17.15	**16.95**	17.05	17.20	17.15
	Q	15.75	15.90	14.90	**14.75**	15.15	14.90	15.15	15.20
	S (28)	13.60	12.15	9.95	9.95	8.90	8.55	**8.00**	8.05
IDA	L	17.70	17.75	**16.70**†	16.90	17.15	17.30	17.10	17.15
	Q	14.85	16.35	14.90	15.05	15.26	**14.65**†	15.15	15.20
	S (28)	12.70	12.05	10.10	10.00	8.90	8.25	**7.85**‡	8.05
PF	L	17.75	17.70	17.10	17.10	**16.95**	17.05	17.20	17.15
	Q	14.90	16.55	14.90	**14.75**	15.15	14.90	15.10	15.20
	S (28)	13.05	12.30	9.85	9.90	8.85	8.55	**8.00**	8.05

TABLE 5
The Estimated Error (Percent) for the Data Set Letter (f)

Method	Classifier	Size of feature space (m)					
		11	12	13	14	15	16 (FULL)
LDA	L	**31.13**†	31.55	31.42	31.32	31.32	31.18
	Q	18.82	17.70	15.62	14.25	12.78	**12.50**†
	S (5)	5.53	4.75	3.90	2.85	**2.12**	2.17
ACC	L	32.20	31.37	31.65	31.57	31.55	**31.18**
	Q	15.38	14.25	13.08	13.12	12.57	**12.50**†
	S (5)	2.67	2.57	2.33	2.23	**2.02**	2.17
IDA	L	32.83	31.60	31.75	31.72	31.37	**31.18**
	Q	15.38	14.10	13.38	13.25	12.65	**12.50**†
	S (5)	3.05	2.33	2.30	2.33	**1.97**‡	2.17
PF	L	32.25	32.15	31.52	31.35	31.45	**31.18**
	Q	16.32	15.10	13.40	12.93	12.70	**12.50**†
	S (5)	3.02	2.62	2.33	2.25	2.20	2.17

is marked by † and the best overall performance is marked by ‡. The performance of the SVM classifier depends on the standard deviation of the Gaussian kernel and the chosen value is shown in the parentheses. To minimize the variability in performance, the feature extraction matrices of the LDA, ACC, and PF methods were orthonormalized prior to classification. This renders all four transformations volume preserving and, so, their performances for a single choice of the kernel are comparable. For generalizable results, both the kernel and the subspace dimension, m, must be estimated from the training data. These could be accomplished, for example, through internal CV over the training sets. In general, this is a computationally expensive procedure and, since the goal of this analysis is to merely compare the performances of various DFE techniques under identical classifiers, no effort was made to optimize the kernel individually for each feature set and/or DFE method.

Based on these results, we make two general observations. First, the SVM classifier uniformly outperforms the quadratic classifier, which is uniformly better than the linear classifier. Second, for both data sets, the best overall performance is based on IDA. In particular, for data set (e), the best results for all three classifiers were observed with the IDA method. For data set (f), however, the results are somewhat mixed, and the performances of the four methods are more comparable. These results are consistent with Section 3.3, where the advantage of IDA was shown to be more substantial on difficult classification problems (measured here by the ratio of $(1 - \mathrm{error})$ and $(1 - \mathrm{chance\ error})$). Also, note that the best error rates per classifier were typically achieved in the subspace of lower dimension than that of the original space, thus underscoring the benefits of the dimensionality reduction. The data sets (e) and (f) were also benchmarked by Torkkola [24], who used a nonparametric mutual information with quadratic entropies as an objective function for finding the optimal DFE matrix. In essence, his technique is equivalent to the PF measure, with the Parzen window estimates of the class-conditional PDFs. While the choice of a Gaussian kernel was not disclosed in [24], the reported optimal error rates with SVMTorch were 10.3 percent for Satellite data and 7.5 percent for Letter data (inferior to the results reported here). Apart from being computationally complex, his method apparently had difficulties with the extraction of high-dimensional subspaces ($m > 15$ for data set (e) and $m > 8$ for data set (f)).

We now turn to the analysis of the remaining data sets, where the k-fold CV was used. The results are summarized in Tables 6, 7, 8, and 9. The errors are shown only for selected subspaces, where at least one method-classifier combination attains a minimum. In addition, standard deviation bounds, estimated from multiple runs of CV, are shown. For a given classifier, the best performance over all DFE methods is marked by †, while ‡ marks the best overall performance. For each classifier-feature space combination, the error estimates that are significantly different (sign test, p = 0.05) from the best error of that combination are given in boldface. For example, the smallest error for the Q-5 combination (Table 6), corresponds to IDA features. This value is significantly different from that of ACC features, but not significantly different from the error corresponding to PF features.

4.2.1 General Performance

As with the previous data sets, we note that the quadratic classifier is generally superior to the linear classifier. Also, note that the average error rates over many subspace

TABLE 6
The Estimated Error (Percent) and Standard Deviation (Percent) for Data Set Brain (a)

Method	Classifier	Size of feature space (m)					
		1	3	4	5	6	8 (FULL)
LDA	L	29.25±1.31	–	–	–	–	29.25±1.31
	Q	28.75±1.51	–	–	–	–	28.50±1.48
ACC	L	**32.50**±2.50	32.12±2.60	30.69±1.95	30.12±2.34	29.38±2.25	29.25±1.31
	Q	**33.06**±2.66	**29.00**±1.70	**28.69**±1.62	**29.12**±1.90	**29.94**±1.64	28.50±1.48
IDA	L	29.19† ±0.79	31.00±0.89	29.56±1.56	29.31±1.29	29.38±1.25	29.25±1.31
	Q	27.50±1.40	25.94‡ ±1.09	26.50±1.88	26.69±0.79	28.81±2.19	**28.50**±1.48
PF	L	30.81±2.41	31.06±0.89	30.38±1.61	29.56±1.81	29.44±1.71	29.25±1.31
	Q	**30.19**±2.81	26.56±1.46	**27.88**±1.68	26.81±0.65	**29.44**±2.04	28.50±1.48

TABLE 7
The Estimated Error (Percent) and Standard Deviation (Percent) for Heart Data (b)

Method	Classifier	Size of feature space (m)				
		1	2	10	11	13 (FULL)
LDA	L	**15.86**±0.62	–	–	–	15.86±0.62
	Q	**16.17**±0.39	–	–	–	17.31±1.22
ACC	L	**17.52**±1.44	**16.79**±1.25	16.03±0.89	15.72±0.90	15.86±0.62
	Q	**17.72**±1.57	18.41±1.49	17.03±0.76	17.28±1.06	17.31±1.22
IDA	L	15.14±0.45	15.10‡±0.53	16.07±0.38	15.90±0.50	**15.86**±0.62
	Q	15.28†±0.44	18.34±0.84	16.76±1.05	17.31±1.12	**17.31**±1.22
PF	L	**16.05**±0.79	**16.02**±0.75	15.97±0.54	15.90±0.59	15.86±0.62
	Q	**16.60**±0.64	18.03±0.85	16.69±1.06	17.28±1.25	17.31±1.22

dimensions remain smaller than those in the full space, thus confirming that a gain in performance can be achieved by reducing the dimensionality of the problem (see Section 2).

4.2.2 Peak Performance

The average error rates of the IDA method compare favorably to those of other techniques for many subspace dimensions m. As indicated earlier, this advantage seems to correlate with the difficulty of the classification problem. In particular, for data set (a), with the achieved error rates of nearly 30 percent, IDA is uniformly (over all m) superior to other DFE methods. Also, note that many of these performance differences are statistically significant, especially for the quadratic classifier. In addition, the best linear and quadratic classifier error rates are those of IDA, with the best overall performance significantly different from the best performances of other techniques. The only exception is the PF method, which, as predicted by our analysis in Section 3.3, approaches the performance of the IDA method when the classes are highly overlapped. A similar analysis applies to other data sets, where the advantages of IDA persist but are not as convincing as in the case of Brain data. Clearly, the class separability in data set (b) is much better than the separability in data set (a), as can be seen by comparing the attained error rates. Nevertheless, the differences in the best error rates of the IDA method and the other three techniques are still statistically significant. Note that the optimal subspace dimensions for data sets (a) and (b) are relatively low $m \in [1, 3]$, consistent with a small number of classes in these data. The performance of IDA on data set (c) is comparable to that of other techniques, most notably due to a high separability of classes, although IDA slightly outperforms other techniques under the linear classifier. Note that the optimal subspace dimension is $m = 1$, regardless of the classifier choice. Finally, the performance of IDA on data set (d) is relatively poor for low-dimensional feature space, especially when compared to LDA, but the performance improves in higher dimensions. In particular, the best linear and quadratic classifier errors are those of IDA. While the best linear classifier result is significantly better than those of other methods, the best quadratic classifier performance is superior to the LDA method only. Note that the performance of LDA is seriously limited by the constraint $m \leq c - 1$.

TABLE 8
The Estimated Error (Percent) and the Standard Deviation (Percent) for Balance Data (c)

Method	Classifier	Size of feature space (m)	
		1	4 (FULL)
LDA	L	**12.97**±0.37	12.97±0.37
	Q	8.40‡±0.13	8.43±0.17
ACC	L	**12.87**±0.31	12.97±0.37
	Q	8.40‡±0.13	8.43±0.17
IDA	L	12.13†±0.56	**12.97**±0.37
	Q	8.40‡±0.13	8.43±0.17
PF	L	**12.90**±0.25	12.97±0.37
	Q	8.40‡±0.13	8.43±0.17

4.2.3 Statistical Significance

To compare the performances of two techniques on certain data, statistical tests are necessary to establish the significance of results. To gather the sufficient statistics for small data sets, data is typically resampled (e.g., holdout, cross-validation, bootstrap [33]). The major problem with this data recycling is that observations violate the independence assumption necessary for further statistical tests. As a consequence, the mere repetition of a k-fold CV will render the performances of any two methods statistically significant (type I error). On the other hand, variability arising from resampling of small data sets [35] may impose differences between two methods that are not genuine (type II error). Dietterich [35] developed a statistical test for a classifier comparison by balancing the two types of errors. When applied to our data, this statistical test produced rather inconsistent results, presumably due to a violation of the many assumptions required by the test. Therefore, a simple sign test on the error samples was performed. Based on these remarks, it should be clear that caution must be exercised when discussing the statistical significance of the results in Section 4.2.2.

TABLE 9
The Estimated Error (Percent) and Standard Deviation (Percent) for Vehicle Data (d)

Method	Classifier	Size of feature space (m)					
		3	10	14	15	16	18 (FULL)
LDA	L	22.08±0.61	–	–	–		22.08±0.61
	Q	20.90±0.71	–	–	–		14.36±0.42
ACC	L	**25.46**±0.39	21.51±0.79	22.44±0.54	22.33±0.66	**21.92**±0.44	22.08±0.61
	Q	**24.79**±0.71	15.49±0.81	13.92±0.73	13.90±0.76	14.05±0.58	14.36±0.42
IDA	L	**32.05**±0.59	**23.26**±0.46	22.51±0.90	22.46±0.80	$21.23^{\dagger}$±0.45	**22.08**±0.61
	Q	**24.26**±0.27	15.59±0.48	$13.59^{\ddagger}$±0.76	13.82±0.80	14.00±0.56	**14.36**±0.42
PF	L	**26.87**±0.43	**22.26**±0.64	21.92±0.87	22.38±0.54	**21.85** ±0.38	22.08±0.61
	Q	**24.85**±0.53	15.62±0.30	13.72±0.77	13.79±0.56	14.18±0.65	14.36±0.42

5 DISCUSSION

We discuss several points related to the implementation and performance of the IDA method.

5.1 Optimization

Both the gradient (15) and the Hessian (16) can be calculated analytically, thus the maximization of the μ-measure is amenable to fast optimization techniques based on Newton's method, such as the trust-region technique [36]. All optimization routines were implemented with the MATLAB™ Optimization Toolbox. The convergence rates of the trust-region method varied across the data sets, but single iterations were generally very fast (a fraction of a second to a couple of seconds), even for problems with the dimension of several hundreds. Supplying the Hessian speeds up optimization procedure by an order of a magnitude. Since the Hessian calculation involves a manipulation of matrices in $\mathbb{R}^{mn \times mn}$, the computation speed will necessarily saturate for large-scale problems. It was observed empirically that, when $mn > 1,000$, a standard conjugate gradient method was a faster optimization scheme, even though it does not facilitate the curvature information. Repeated optimization runs with 10 randomized initial conditions yielded solutions with less than 10^{-8} percent of relative improvement over a single optimization run. These improvements are comparable to the relative tolerance of the optimization routine and suggest that the optimal results were independent of the choice of initial condition. Experiments with simulated annealing [37] using an exponential annealing schedule were not able to improve upon the best solution obtained through random restarts; therefore, the maximization of the μ-measure does not appear prone to the problem of local maxima, at least for data studied in this paper. Choosing a good initial condition, however, speeds up the overall computation considerably. The feature extraction matrix of the ACC method and, when admissible, the LDA method, were used as an initial guess for the optimization problem (14). In many cases, the μ-measure in the LDA and ACC feature subspace was very close to the optimal μ-measure found by IDA (< 1 percent in some cases). Thus, LDA and ACC provide a good initial condition for the maximization of μ and, in turn, its fast convergence. A similar initialization approach was used in [24].

5.2 Singular Covariance Matrices

Since second-order techniques involve covariance matrices, problems may arise if these matrices are singular. This, for example, often happens in the so-called small sample size problems [1, p. 39], such as image classification, where the dimension of data, n, exceeds the number of data instances. As this is a problem commonly faced by many second-order techniques, including LDA, there have been a number of proposed solutions, ranging from the removal of data singularities through PCA [17], to various shrinkage approaches [38]. Some recent solutions to the small sample size problem in the context of LDA include various covariance matrix subspace decompositions [39], [40] and combining the discriminatory information over the subspaces.

The μ-measure (8) becomes ill-defined if any of the covariance matrices Σ_i are singular. However, one benefit of IDA, over LDA and ACC in particular, is that it does not deal with covariance matrices Σ_i directly, rather, it relies on their feature space representation $\boldsymbol{T}\,\Sigma_i\,\boldsymbol{T}^{\mathrm{T}}$. It follows immediately that $\mathrm{rank}(\boldsymbol{T}\,\Sigma_i\,\boldsymbol{T}^{\mathrm{T}}) = \min\{m, r_i\}$, where r_i is the rank of Σ_i. Therefore, as long as the size of the feature space is smaller than the rank of Σ_i, the μ-measure and its subsequent optimization will be well-defined. In the small sample size problems, r_i is typically linked to the number of training instances, n_i, in class ω_i, i.e., $r_i = n_i - 1$, while m is small by design and, so, the assumption $m < r_i$ is well-justified. Extending m beyond r_i would require the use of aforementioned techniques, such as covariance shrinkage. In addition, subspace decompositions, proposed in [39], [40], may provide some computational savings by removing large uninformative subspaces.

5.3 Performance

In general, the parametric form (8) of the μ-measure can be viewed as an approximation of the μ-measure (6) when class-conditional PDFs are replaced by their second order approximations and, so, the μ-measure (8) is ideally suited for Gaussian classes. While Fig. 1 and extensive Monte Carlo simulations (Section 3.3) indicate that IDA works well for non-Gaussian classes, we also comment on the implications of the second order approximation on experimental data. Consider, for example, data sets (e) and (f), where classes are known to deviate from the Gaussian assumption [24], and consider the performance of the SVM classifier in the full space (Table 4 and Table 5), which is Gaussian assumption-free. Evidently, the performances are comparable (even inferior) to the peak performance of IDA and other second-order techniques. This suggests that the violation of the Gaussian assumption is not critical for the performance of IDA. In cases where classes deviate significantly from the Gaussian assumption, IDA (and any other second-order technique) may yield suboptimal solutions; though as long as the first two statistical moments contain discriminatory information, IDA is expected to perform well.

Although the number of analyzed data sets in this study is limited (9), data comes from a variety of domains, with a diverse number of classes, attributes, and sample sizes. In addition, many experimental data sets contain a combination of continuous, discrete, and nominal attributes. Because of this data diversity, we hope that the conclusions of this study will hold in a more general set-up. Based on the results presented in Tables 1, 2, 3, 4, 5, 6, 7, 8, and 9, it follows that IDA outperforms other methods according to many criteria such as: the number of best performances, the number of best overall performances and the optimality margin. Even when suboptimal, the performance of the IDA method remains relatively close to the optimal performance. We conclude that, among the DFE methods tested on the nine data sets, IDA is the best single technique.

Clearly, the IDA technique is computationally more demanding than the ACC method and especially the LDA method. Since the absolute improvements of IDA are modest (< 7.5 percent for LDA; < 2.75 percent for ACC), it is worth addressing the question of IDA's justifiability. First note that the μ-measure (8), the gradient (15), and the Hessian (16) are available analytically. Moreover, they involve simple matrix manipulations. The optimization problem (14) can be solved using standard algorithms (trust-region and conjugate gradients), which are readily available. Finally, once the LDA (ACC) feature extraction matrix is available as an initial guess for the IDA matrix, the solution is typically found in only a few iterations. Even if a random initial condition is used, the solution is available within seconds (minutes for large-scale problems). This is not the case for the PF method and related probabilistic dependence measures [14], which are computationally much more intense, even when the parametric class models are assumed. While available analytically, the expression for the PF measure involves $\mathcal{O}(c^3)$ terms. Thus, PF will be very slow for problems involving a large number of classes (e.g., data set (f)). Similarly, the expression for the gradient is equally complex and there are no known expressions for the Hessian, which limit the practical applicability of the PF measure. In summary, IDA offers a computationally feasible alternative to other linear DFE methods.

6 Conclusion

Using elementary information-theoretic tools, we have developed a novel linear DFE method, conveniently called IDA. The method facilitates the maximization of a measure, μ, which, under the parametric class-conditional PDF models, can be analytically computed from the data. We have shown that the μ-measure has many interesting properties that are reminiscent of the mutual information and the Bayes error.

If the classes conform to the homoscedastic Gaussian conditions, IDA reduces to the classical LDA technique and is an optimal feature extraction technique in the sense of Bayes. For two closely centered heteroscedastic Gaussian classes, the μ-measure reduces to the Chernoff distance. Sufficient conditions for the optimality of IDA in the sense of Bayes have been given for heteroscedastic Gaussian classes. We have justified the use of the μ-measure as a class-separability criterion by showing how it relates to the differences in the class-conditional means and the class-conditional covariances, which, in turn, makes IDA suitable for heteroscedastic data.

Finally, we have tested the performance of the IDA method, and several related second-order techniques, on a number of simulated and real-world data sets. We have demonstrated, both theoretically and experimentally, that, when class-conditional PDFs are highly overlapped, the IDA method outperforms other second-order techniques, and as the classes are more separated, the performance of IDA approaches that of the other methods. Since the estimation of the IDA feature extraction matrix is computationally feasible, IDA should be considered as an alternative to other linear DFE methods.

Acknowledgments

The author would like to thank Mathias Jungen for reading an earlier version of the manuscript and for his valuable comments. The author would also like to acknowledge the anonymous reviewers for their constructive criticism and the editorial staff of this journal.

References

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition,* second ed. Academic Press, 1990.

[2] P.J. Huber, "Projection Pursuit," *Ann. Statistics,* vol. 13, no. 2, pp. 435-475, 1985.

[3] J.H. Friedman and J.W. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis," *IEEE Trans. Computers,* vol. 23, pp. 881-889, 1974.

[4] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, pp. 2323-2326, 2000.

[5] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, pp. 2319-2323, 2000.

[6] I.T. Jolliffe, *Principal Component Analysis.* Springer Verlag, 1986.

[7] A. Hyvärinen, "Survey on Independent Component Analysis," *Neural Computing Surveys,* vol. 2, pp. 94-128, 1999.

[8] R.A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics,* vol. 7, pp. 179-188, 1936.

[9] C.R. Rao, "The Utilization of Multiple Measurements in Problems of Biological Classification," *J. Royal Statistical Soc. B,* vol. 10, no. 2, pp. 159-203, 1948.

[10] N. Kumar and A.G. Andreou, "Heteroscedastic Discriminant Analysis and Reduced Rank HMMs for Improved Speech Recognition," *Speech Comm.,* vol. 26, pp. 283-297, 1998.

[11] K. Torkkola, "Discriminative Features for Document Classification," *Proc. 16th Int'l Conf. Pattern Recognition,* vol. 1, pp. 472-475, 2002.

[12] L.-F. Chen, H.-Y.M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "A New LDA-Based Face Recognition System which Can Solve the Small Sample Size Problem," *Pattern Recognition,* vol. 33, no. 10, pp. 1713-1726, 2000.

[13] G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition.* Wiley, 1992.

[14] P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Prentice Hall, 1982.

[15] H.P. Decell and J.A. Quirein, “An Iterative Approach to the Feature Selection Problem,” *Proc. Purdue Conf. Machine Processing of Remotely Sensed Data*, pp. 3B1-3B12, 1972.

[16] G. Saon and M. Padmanabhan, “Minimum Bayes Error Feature Selection for Continuous Speech Recognition,” *Advances in Neural Information Processing Systems 13,* pp. 800-806, 2001.

[17] M. Loog and R.P.W. Duin, “Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Chernoff Criterion,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, pp. 732-739, 2004.

[18] P.M. Lewis II, “The Characteristic Selection Problem in Recognition Systems,” *IEEE Trans. Information Theory*, vol. 8, no. 2, pp. 171-178, 1962.

[19] R. Battiti, “Using Mutual Information for Selecting Features in Supervised Neural Net Learning,” *IEEE Trans. Neural Networks,* vol. 5, no. 4, pp. 537-550, 1994.

[20] J. Kittler, “Feature Set Search Algorithms,” *Pattern Recognition and Signal Processing,* pp. 41-60, 1978.

[21] T.M. Cover and J.M. Van Campenhout, “On the Possible Ordering in the Measurement Selection Problem,” *IEEE Trans. Systems, Man, and Cybernetics,* vol. 7, pp. 657-661, 1977.

[22] J.C. Principe, J.W. Fisher III, and D. Xu, “Information Theoretic Learning,” *Unsupervised Adaptive Filtering,* 2000.

[23] A. Renyi, “On Measures of Entropy and Information,” *Proc. Fourth Berkeley Symp. Math. Statistics and Probability,* pp. 547-561, 1961.

[24] K. Torkkola, “Feature Extraction by Nonparamatric Mutual Information Maximization,” *J. Machine Learning Research,* vol. 3, pp. 1415-1438, 2003.

[25] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* Wiley-Interscience, 2001.

[26] M.E. Hellman and J. Raviv, “Probability of Error, Equivocation, and the Chernoff Bound,” *IEEE Trans. Information Theory,* vol. 16, no. 4, pp. 368-372, 1970.

[27] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* Wiley Interscience, 1991.

[28] M.C. Jones and R. Sibson, “What Is Projection Pursuit?” *J. Royal Statistical Soc., Series A,* vol. 150, pp. 1-36, 1987.

[29] D.H. Johnson and D.E. Dudgeon, *Array Signal Processing: Concepts and Techniques.* Prentice Hall, 1993.

[30] S. Hettich, C.L. Blake, and C.J. Merz, “UCI Repository of Machine Learning Databases,” Dept. of Information and Computer Sciences, Univ. of California, Irvine, http://www.ics.uci.edu/mlearn/MLRepository.html, 1998.

[31] D.S. Rizzuto, A.N. Mamelak, W.W. Sutherling, I. Fineman, and R.A. Andersen, “Spatial Selectivity in Human Ventrolateral Prefrontal Cortex,” *Nature Neuroscience,* vol. 8, pp. 415-417, 2005.

[32] R. Collobert and S. Begnio, “SVMTorch: Support Vector Machines for Large-Scale Regression Problems,” *J. Machine Learning Research,* vol. 1, no. 2, pp. 143-160, 2001.

[33] R. Kohavi, “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection,” *Proc. Int'l Joint Conf. Artifical Intelligence,* pp. 1137-1145, 1995.

[34] B. Effron, “Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation,” *J. Am. Statistical Assoc.*, 1983.

[35] T.G. Dietterich, “Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms,” *Neural Computation,* vol. 10, pp. 1895-1923, 1998.

[36] J. Nocedal and S.J. Wright, *Numerical Optimization.* Springer, 1999.

[37] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, “Optimization by Simulated Annealing,” *Science,* vol. 220, no. 4598, pp. 671-680, 1983.

[38] J. Schäfer and K. Strimmer, “A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics,” *Statistical Applications in Genetics and Molecular Biology,* vol. 4, no. 1, p. 32, 2005.

[39] H. Yu and H. Yang, “A Direct LDA Algorithm for High-Dimensional Data—With Application to Face Recognition,” *Pattern Recognition Letters,* vol. 34, no. 10, pp. 2067-2070, 2001.

[40] X. Wang and X. Tang, “Dual-Space Linear Discriminant Analysis for Face Recognition,” *Proc. 2004 IEEE Conf. Computer Vision and Pattern Recognition,* vol. 22, pp. 564-569, 2004.

[41] G. Visick, “A Quantitative Version of the Observation that the Hadamard Product Is a Principal Submatrix of the Kronecker Product,” *Linear Algebra Appliction,* vol. 304, pp. 45-68, 2000.

[42] P.S. Dwyer, “Some Applications of Matrix Derivatives in Multivariate Analysis,” *J. Am. Statistical Assoc.,* vol. 62, no. 3, pp. 607-625, 1967.

[43] M. Brookes, “The Matrix Reference Manual,” Imperial College, London, http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/calculus.html, 2005.

[44] R.A. Horn and C.R. Johnson, *Matrix Analysis.* Cambridge Univ. Press, 1985.

[45] C.R. Johnson, “Partitioned and Hadamard Product Matrix Inequalities,” *J. Research Nat'l Bureau of Standards,* vol. 83, pp. 585-591, 1978.

[46] A.V. Balakrishnan, *Kalman Filtering Theory.* Optimization Software Inc., 1987.

[47] M. Padmanabhan and S. Dharanipragada, “Maximizing Information Content in Feature Extraction,” *IEEE Trans. Speech Audio Processing,* vol. 13, no. 4, pp. 512-519, 2005.

Zoran Nenadic received the diploma in control engineering from the University of Belgrade, Serbia, in 1995 and the MS and DSc degrees in systems science and mathematics from Washington University, St. Louis, Missouri, in 1998 and 2001, respectively. From 2001 to 2005, he was a postdoctoral fellow with the Division of Engineering and Applied Science at the California Institute of Technology, Pasadena. Since 2005, he has been with the Department of Biomedical Engineering, University of California, Irvine, where he is currently an assistant professor. His research interests are in the area of adaptive biomedical signal processing, control algorithms for biomedical devices, brain-machine interfaces, and modeling and analysis of biological neural networks. He is a member of the IEEE, the Mathematical Association of America, and the Society for Neuroscience.

APPENDIX I
PROOF OF PROPOSITION 1

It follows from (8) that

$$\begin{aligned}\mu(\mathsf{R};\Omega) &= \frac{1}{2}\left[\log\left(\left|\boldsymbol{\Sigma}_{\mathrm{W}} + \sum_{i=1}^{c} \tilde{\boldsymbol{m}}_i \tilde{\boldsymbol{m}}_i{}^{\mathrm{T}} p_i\right|\right)\right.\\ &\left. - \sum_{i=1}^{c} \log(|\boldsymbol{\Sigma}_i|) p_i\right]\end{aligned}$$

where $\boldsymbol{\Sigma}_{\mathrm{W}} \triangleq \sum_{i=1}^{c} \boldsymbol{\Sigma}_i \, p_i$. Based on (22) (see Appendix III), followed by the chain rule, we obtain

$$\frac{\partial \mu}{\partial \tilde{\boldsymbol{m}}_j} = p_j \left[\boldsymbol{\Sigma}_{\mathrm{W}} + \sum_{i=1}^{c} \tilde{\boldsymbol{m}}_i \tilde{\boldsymbol{m}}_i{}^{\mathrm{T}} p_i\right]^{-1} \tilde{\boldsymbol{m}}_j$$

To prove (10), it suffices to show that $\left[\boldsymbol{\Sigma}_{\mathrm{W}} + \sum_{i=1}^{c} \tilde{\boldsymbol{m}}_i \, \tilde{\boldsymbol{m}}_i{}^{\mathrm{T}} p_i\right]^{-1}$ is a positive definite matrix, which follows readily after noting that $\boldsymbol{\Sigma}_{\mathrm{W}} > 0$ and $\left[\sum_{i=1}^{c} \tilde{\boldsymbol{m}}_i \, \tilde{\boldsymbol{m}}_i{}^{\mathrm{T}} p_i\right] > 0$. ■

APPENDIX II
PROOF OF PROPOSITION 2

Based on (8) and (21) we calculate $\partial\mu/\partial\tilde{\boldsymbol{\Sigma}}_j$ as

$$\frac{\partial \mu}{\partial \tilde{\boldsymbol{\Sigma}}_j} = \frac{p_j}{2}\left[\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}_j^{-1}\right]$$

To prove (11) we need to show $\langle \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}_j^{-1} : \boldsymbol{\Sigma}_j - \boldsymbol{\Sigma}\rangle > 0$. This condition can be written as

$$\boldsymbol{e}^{\mathrm{T}}\left[\left(\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}_j^{-1}\right) \circ (\boldsymbol{\Sigma}_j - \boldsymbol{\Sigma})\right] \boldsymbol{e} > 0$$

where $\boldsymbol{e} \triangleq [1,\, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^n$, and $\circ$ denotes the Hadamard product (see Appendix IV). Based on the properties 2) and 5) of the Hadamard product (see Appendix IV), the expression above reduces to[8]

$$\boldsymbol{e}^{\mathrm{T}}\left(\boldsymbol{\Sigma}^{-1} \circ \boldsymbol{\Sigma}_j + \boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma}\right) \boldsymbol{e} > 2\, n \tag{17}$$

[8] A less strict version of the condition (17), written as $\boldsymbol{\Sigma}^{-1} \circ \boldsymbol{\Sigma}_j + \boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma} \geq 2\boldsymbol{I}$, was proven in [41].

Based on the property 6), the term on the left hand side of (17), denoted by L, satisfies the following

$$\begin{aligned}
&\text{case (1): } \mathrm{L} = \boldsymbol{e}^{\mathrm{T}} \left[\left(\boldsymbol{\Sigma} \circ \boldsymbol{\Sigma}_j^{-1} \right)^{-1} + \boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma} \right] \boldsymbol{e} && (18)\\
&\boldsymbol{\Sigma}_j, \boldsymbol{\Sigma} - \text{diagonal} &&\\
&\text{case (2): } \mathrm{L} > \boldsymbol{e}^{\mathrm{T}} \left[\left(\boldsymbol{\Sigma} \circ \boldsymbol{\Sigma}_j^{-1} \right)^{-1} + \boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma} \right] \boldsymbol{e} && (19)\\
&\ \text{otherwise} &&
\end{aligned}$$

We proceed by noting that the matrix $\left(\boldsymbol{\Sigma} \circ \boldsymbol{\Sigma}_j^{-1}\right)^{-1} + \boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma}$ is symmetric [property 3)], therefore its eigenvectors $\{\boldsymbol{v}_i : i = 1, 2, \cdots, n\}$ form an orthonormal basis in $\mathbb{R}^n$. Write $\boldsymbol{e}$ in this new basis as $\boldsymbol{e} = \sum_{i=1}^n g_i \boldsymbol{v}_i$, where $g_i \triangleq \langle \boldsymbol{e}, \boldsymbol{v}_i \rangle$, and observe that $\boldsymbol{e}^{\mathrm{T}}[\left(\boldsymbol{\Sigma} \circ \boldsymbol{\Sigma}_j^{-1}\right)^{-1} + \boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma}]\boldsymbol{e} = \sum_{i=1}^n g_i^2 \alpha_i$, where α_i are the eigenvalues corresponding to the eigenvectors $\boldsymbol{v}_i$. Based on the property 1) and the properties of invertible matrices we note that $\alpha_i = \lambda_i + 1/\lambda_i$, where λ_i are the eigenvalues of $\boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma}$. Finally, observe that $\boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma}$ is a positive definite matrix [property 4)], therefore $\lambda_i > 0$, $\forall i$. Thus $\min_{\lambda > 0} \{\lambda + 1/\lambda\} = 2$, and the minimum is attained at $\lambda = 1$. Based on these observations we have

$$\begin{aligned}
&\boldsymbol{e}^{\mathrm{T}} \left[\left(\boldsymbol{\Sigma} \circ \boldsymbol{\Sigma}_j^{-1} \right)^{-1} + \boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma} \right] \boldsymbol{e} \\
&\qquad\qquad \geq 2 \sum_{i=1}^n g_i^2 = 2\, n \qquad (20)
\end{aligned}$$

and the equality holds if and only if $\boldsymbol{e}$ is the eigenvector of $\boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma}$ corresponding to $\lambda = 1$. Therefore, for the case (2) the condition (17) follows directly from (19) and (20). A similar argument applies to the case (1), the only concern arising when (20) holds with equality. But in this case, the condition $\left(\boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma}\right) \boldsymbol{e} = \boldsymbol{e}$ imposes that the row sums of the matrix $\boldsymbol{\Sigma}_j^{-1} \circ \boldsymbol{\Sigma}$ are 1, which for diagonal matrices $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\Sigma}$ yields $\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}$. This, however, contradicts our assumption $\tilde{\boldsymbol{\Sigma}}_j \neq \boldsymbol{0}_n$. ■

Appendix III

Derivatives of Functions of Matrices

Let $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ be a differentiable function of a matrix $\boldsymbol{T} \in \mathbb{R}^{m \times n}$. We define $\partial f / \partial \boldsymbol{T}$ as an $m \times n$ matrix such that $[\partial f / \partial \boldsymbol{T}]_{i,j} \triangleq \partial f / \partial T_{i,j}$. Let $\boldsymbol{\Psi} : \mathbb{R}^{m \times n} \to \mathbb{R}^{p \times q}$ be a differentiable function of a matrix $\boldsymbol{T} \in \mathbb{R}^{m \times n}$. We define $\partial \boldsymbol{\Psi} / \partial \boldsymbol{T}$ as a $pq \times mn$ matrix such that

$[\partial\boldsymbol{\Psi}/\partial\boldsymbol{T}]_{(j-1)p+i,(l-1)m+k} \triangleq \partial\Psi_{i,j}/\partial T_{k,l}$. We list some important rules and identities for the differentiation of functions of matrices [42], [43]. Unless otherwise noted, it is assumed that $\boldsymbol{T} \in \mathbb{R}^{m\times n}$, $\boldsymbol{\Psi} : \mathbb{R}^{r\times s} \to \mathbb{R}^{p\times q}$ and $\boldsymbol{\Phi} : \mathbb{R}^{m\times n} \to \mathbb{R}^{r\times s}$ are differentiable functions of their respective variables.

1) $\partial|\boldsymbol{T}|/\partial\boldsymbol{T} = \mathrm{adj}^{\mathrm{T}}(\boldsymbol{T})$, where $\boldsymbol{T} \in \mathbb{R}^{n\times n}$.
2) $\partial\boldsymbol{T}/\partial\boldsymbol{T} = \boldsymbol{I}_n \otimes \boldsymbol{I}_m$, where $\otimes$ stands for the Kronecker product [44].
3) $\partial\boldsymbol{T}^{-1}/\partial\boldsymbol{T} = -\left(\boldsymbol{T}^{-\mathrm{T}} \otimes \boldsymbol{T}^{-1}\right)$, where $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ is a non-singular matrix.
4) Product rule: $\partial\left[\boldsymbol{U}\boldsymbol{V}\right]/\partial\boldsymbol{T} = (\boldsymbol{I}_r \otimes \boldsymbol{U})\,\partial\boldsymbol{V}/\partial\boldsymbol{T} + (\boldsymbol{V}^{\mathrm{T}} \otimes \boldsymbol{I}_p)\,\partial\boldsymbol{U}/\partial\boldsymbol{T}$, where $\boldsymbol{U} : \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$ and $\boldsymbol{V} : \mathbb{R}^{m\times n} \to \mathbb{R}^{q\times r}$ are differentiable functions of $\boldsymbol{T}$.
5) Chain rule: $\partial\left[\boldsymbol{\Psi}\left(\boldsymbol{\Phi}(\boldsymbol{T})\right)\right]/\partial\boldsymbol{T} = \left[\partial\boldsymbol{\Psi}/\partial\boldsymbol{\Phi}\right]\left[\partial\boldsymbol{\Phi}(\boldsymbol{T})/\partial\boldsymbol{T}\right]$.

From the foregoing, we have the following useful results:

- If $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ is an invertible matrix with $|\boldsymbol{T}| > 0$, then
$$\frac{\partial\log(|\boldsymbol{T}|)}{\partial\boldsymbol{T}} = \boldsymbol{T}^{-\mathrm{T}} \tag{21}$$
- Let $\boldsymbol{T} \in \mathbb{R}^{m\times n}$ $(m < n)$ be a full-rank matrix, and let $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. From (21) and the chain rule we have
$$\frac{\partial\log(|\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^{\mathrm{T}}|)}{\partial\boldsymbol{T}} = 2(\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^{\mathrm{T}})^{-1}\boldsymbol{T}\boldsymbol{\Sigma} \tag{22}$$
- Similarly, based on the product rule and the chain rule we write
$$\begin{aligned}\frac{\partial\left(\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^{\mathrm{T}}\right)}{\partial\boldsymbol{T}} &= (\boldsymbol{I}_m \otimes \boldsymbol{T}\boldsymbol{\Sigma})\frac{\partial\boldsymbol{T}^{\mathrm{T}}}{\partial\boldsymbol{T}}\\ &+ (\boldsymbol{T} \otimes \boldsymbol{I}_m)\frac{\partial(\boldsymbol{T}\boldsymbol{\Sigma})}{\partial\boldsymbol{T}}\\ &= (\boldsymbol{I}_m \otimes \boldsymbol{T}\boldsymbol{\Sigma})\,\boldsymbol{\Theta}\\ &+ (\boldsymbol{T}\boldsymbol{\Sigma} \otimes \boldsymbol{I}_m) \triangleq \boldsymbol{C}(\boldsymbol{\Sigma})\end{aligned}$$
where we have used the fact that $(\boldsymbol{T} \otimes \boldsymbol{I}_m)(\boldsymbol{\Sigma} \otimes \boldsymbol{I}_m) = (\boldsymbol{T}\boldsymbol{\Sigma} \otimes \boldsymbol{I}_m)$. The matrix $\boldsymbol{\Theta} \in \mathbb{R}^{mn\times mn}$ is a sparse matrix with $\{\Theta_{(j-1)n+i,(i-1)m+j} = 1 \,|\, i = 1,\cdots,n;\ j = 1,\cdots,m\}$ and all other elements 0.
- Finally, based on the product rule we find the derivative of the main term in (22) as
$$\frac{\partial\left[\left(\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^{\mathrm{T}}\right)^{-1}\boldsymbol{T}\boldsymbol{\Sigma}\right]}{\partial\boldsymbol{T}} = \boldsymbol{A}(\Sigma) + \boldsymbol{B}(\Sigma)$$

where from the chain rule we have

$$\begin{aligned}\boldsymbol{A}(\boldsymbol{\Sigma}) &= \left[\boldsymbol{I}_n \otimes \left(\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^{\mathrm{T}}\right)^{-1}\right](\boldsymbol{\Sigma}\otimes \boldsymbol{I}_m)\\ \boldsymbol{B}(\boldsymbol{\Sigma}) &= -\left(\boldsymbol{\Sigma}\boldsymbol{T}^{\mathrm{T}} \otimes \boldsymbol{I}_m\right)\\ &\quad \left[\left(\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^{\mathrm{T}}\right)^{-\mathrm{T}} \otimes \left(\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^{\mathrm{T}}\right)^{-1}\right]\boldsymbol{C}(\boldsymbol{\Sigma})\end{aligned} \tag{23}$$

where $\boldsymbol{A}(\boldsymbol{\Sigma}), \boldsymbol{B}(\boldsymbol{\Sigma}) \in \mathbb{R}^{mn\times mn}$.

APPENDIX IV
SOME PROPERTIES OF HADAMARD PRODUCT

The Hadamard (elementwise) product of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same size is defined as $[\boldsymbol{A}\circ\boldsymbol{B}]_{i,j} \triangleq A_{i,j}\, B_{i,j}$. We will use the following properties of the Hadamard product:

1) Commutativity: $\boldsymbol{A}\circ\boldsymbol{B} = \boldsymbol{B}\circ\boldsymbol{A}$ (follows directly from the definition).
2) Distributivity: $\boldsymbol{A}\circ(\boldsymbol{B}+\boldsymbol{C}) = \boldsymbol{A}\circ\boldsymbol{B} + \boldsymbol{A}\circ\boldsymbol{C}$ (follows from the definition).
3) If $\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}$ and $\boldsymbol{B} = \boldsymbol{B}^{\mathrm{T}}$, then $(\boldsymbol{A}\circ\boldsymbol{B})^{\mathrm{T}} = \boldsymbol{A}^{\mathrm{T}}\circ\boldsymbol{B}^{\mathrm{T}} = \boldsymbol{A}\circ\boldsymbol{B}$ (follows from the definition).
4) If $\boldsymbol{A} > 0$ and $\boldsymbol{B} > 0$, then $(\boldsymbol{A}\circ\boldsymbol{B}) > 0$ (see [44, pp. 458]).
5) If $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ is an invertible matrix and $\boldsymbol{e} \triangleq [1,\, 1,\, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^n$, then $\boldsymbol{e}^{\mathrm{T}}\left(\boldsymbol{A}^{-1}\circ\boldsymbol{A}\right)\boldsymbol{e} = n$ (follows from the definition, but see also [45]).
6) If $\boldsymbol{A} > 0$ and $\boldsymbol{B} > 0$, then $(\boldsymbol{A}\circ\boldsymbol{B})^{-1} \leq \boldsymbol{A}^{-1}\circ\boldsymbol{B}^{-1}$ and the equality holds if and only if $\boldsymbol{A}$ and $\boldsymbol{B}$ are both diagonal (see [45]).

APPENDIX V
PROOF OF COROLLARY 1

It suffices to show that the conditions of Corollary 1 imply the conditions of Theorem 4, i.e. $f_{\mathsf{N|S},\Omega}(\boldsymbol{n}\,|\,\boldsymbol{s},\omega_i) = f_{\mathsf{N|S}}(\boldsymbol{n}\,|\,\boldsymbol{s})$, $\forall i = \{1, 2, \cdots, c\}$, $\forall \boldsymbol{s}\in\mathbb{R}^m$, $\forall \boldsymbol{n}\in\mathbb{R}^d$. Since $\mathsf{S}|\Omega$ and $\mathsf{N}|\Omega$ are Gaussian, so is $\mathsf{N}|\mathsf{S},\Omega$, i.e. $\mathsf{N}|\mathsf{S},\Omega \sim \mathcal{N}\left(\boldsymbol{m}_i^{\mathsf{N|S}}, \boldsymbol{\Sigma}_i^{\mathsf{N|S}}\right)$, where [46, pp. 45-51]

$$\begin{aligned}\boldsymbol{m}_i^{\mathsf{N|S}} &= \boldsymbol{m}_i^{\mathsf{N}} + \boldsymbol{\Sigma}_i^{\mathsf{NS}}(\boldsymbol{\Sigma}_i^{\mathsf{SS}})^{-1}\left(\boldsymbol{s}-\boldsymbol{m}_i^{\mathsf{S}}\right)\\ \boldsymbol{\Sigma}_i^{\mathsf{N|S}} &= \boldsymbol{\Sigma}_i^{\mathsf{NN}} - \boldsymbol{\Sigma}_i^{\mathsf{NS}}(\boldsymbol{\Sigma}_i^{\mathsf{SS}})^{-1}\boldsymbol{\Sigma}_i^{\mathsf{SN}} \qquad \forall i,\ \forall \boldsymbol{s}\end{aligned} \tag{24}$$

After recalling the conditions of Corollary 1, it follows immediately from (24) that $\boldsymbol{m}_i^{\mathsf{N|S}} = \boldsymbol{m}^{\mathsf{N}}$ and $\boldsymbol{\Sigma}_i^{\mathsf{N|S}} = \boldsymbol{\Sigma}^{\mathsf{NN}}$, hence $\mathsf{N}|\mathsf{S},\Omega \sim \mathcal{N}\left(\boldsymbol{m}^{\mathsf{N}}, \boldsymbol{\Sigma}^{\mathsf{NN}}\right)$. By definition

$$f_{\mathsf{N|S}}(\boldsymbol{n}\,|\,\boldsymbol{s}) = \sum_{i=1}^{c} f_{\mathsf{N|S},\Omega}(\boldsymbol{n}\,|\,\boldsymbol{s},\omega_i)\, P(\omega_i|\boldsymbol{s})$$

which after noting that $f_{\mathsf{N}|\mathsf{S},\Omega}(\boldsymbol{n}\,|\,\boldsymbol{s},\omega_i) = \mathcal{N}\left(\boldsymbol{m}^{\mathsf{N}},\boldsymbol{\Sigma}^{\mathsf{NN}}\right)$ and $\sum_{i=1}^{c} P(\omega_i|\boldsymbol{s}) = 1$, implies that $\mathsf{N}|\mathsf{S} \sim \mathcal{N}\left(\boldsymbol{m}^{\mathsf{N}},\boldsymbol{\Sigma}^{\mathsf{NN}}\right)$. ■