Pattern Recognition 48 (2015) 1863–1877

Contents lists available at ScienceDirect

Pattern Recognition

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

Pareto models for discriminative multiclass linear dimensionality reduction

Karim T. Abou-Moustafa [a,*], Fernando De La Torre [b], Frank P. Ferrie [c]

[a] Dept. of Computing Science, ATH 3-55, University of Alberta, Edmonton, AB, Canada T6G 2E8
[b] Robotics Institute, Smith Hall 211, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[c] Dept. of Electrical & Computer Engineering and Centre of Intelligent Machines, McGill University, McConnell Engineering Building, Room 441, 3480 University Street, Montreal, QC, Canada H3A 2E9

ARTICLE INFO

Article history:
Received 10 January 2014
Received in revised form
13 August 2014
Accepted 8 November 2014
Available online 20 November 2014

Keywords:
Fisher discriminant analysis
Supervised linear dimensionality reduction
Feature transformation
Metric learning
Subspace learning
Multiobjective optimization
Pareto optimality
Kullback–Leibler divergence

ABSTRACT

We address the class masking problem in multiclass linear discriminant analysis (LDA). In the multiclass setting, LDA does not maximize each pairwise distance between classes, but rather maximizes the sum of all pairwise distances. This results in serious overlaps between classes that are close to each other in the input space, and degrades classification performance. Our research proposes Pareto Discriminant Analysis (PARDA); an approach for multiclass discriminative analysis that builds over multiobjective optimizing models. PARDA decomposes the multiclass problem to a set of objective functions, each representing the distance between every pair of classes. Unlike existing LDA extensions that maximize the sum of all distances, PARDA maximizes each pairwise distance to maximally separate all class means, while minimizing the class overlap in the lower dimensional space. Experimental results on various data sets show consistent and promising performance of PARDA when compared with well-known multiclass LDA extensions.

1. Introduction

Fisher Discriminant Analysis (FDA) originally developed by Fisher in 1936 [1] is a technique for supervised linear dimensionality reduction that is optimal for classification under two assumptions: (i) the number of classes c is exactly two, and (ii) the samples in each class are assumed to be generated from a multivariate Gaussian distribution with different means and equal covariance matrices (homoscedastic data) [2]. In this context, FDA is guaranteed to find a one dimensional subspace that will classify the samples with the optimal error rate, *Bayes error*, and the subspace is known to be *Bayes optimal* [2]. Rao [3] extended this approach to the multiclass homoscedastic case ($c > 2$), under the condition that the data features $d \geq c$ (and assuming the number of samples $n > d$). The resultant $c-1$ dimensional subspace is also guaranteed to be Bayes optimal, and the technique has become known as Linear Discriminant Analysis (LDA). Rao also noted that in the homoscedastic case, if the lower dimensional subspace has dimensionality $d_0 < c-1$, the resultant subspace will not be Bayes optimal. It is only recently that Hamsici and Martinez [4] pushed the homoscedastic case further and derived a Bayes optimal one dimensional subspace when $c > 2$.

When the equal covariance assumption does not hold for $c \geq 2$ (heteroscedastic data), Rao proposed to approximate the heteroscedastic problem with a homoscedastic setting and solve the approximated problem instead. His approximated problem considered that all classes have different means but share a common covariance matrix which is a weighted average of all the covariance matrices of the original problem. This approximation matrix became known as the pooled sample covariance matrix, or the average within-class scatter matrix $\mathbf{S}_w$. Rao's final solution became the well known formulation based on the Rayleigh quotient of the between-class scatter matrix $\mathbf{S}_b$ and $\mathbf{S}_w$. The obtained subspace, however, is not Bayes optimal for the original heteroscedastic problem.

Several researchers, backed by theoretical justifications, have scrutinized the limitations and non-optimality (in terms of Bayes error) of LDA when its strong assumptions do not hold and proposed extensions derived from Gaussian assumptions [5–8] and kernel methods [9,10] to generalize LDA to the multiclass heteroscedastic case. The result was a plethora of algorithms that have been reported to perform well in a variety of application

* Corresponding author. Tel.: +1 780 655 2685; fax: +1 780 492 1071.
E-mail addresses: aboumous@cs.ualberta.ca (K.T. Abou-Moustafa), ftorre@cs.cmu.edu (F. De La Torre), ferrie@cim.mcgill.ca (F.P. Ferrie).

http://dx.doi.org/10.1016/j.patcog.2014.11.008
0031-3203/

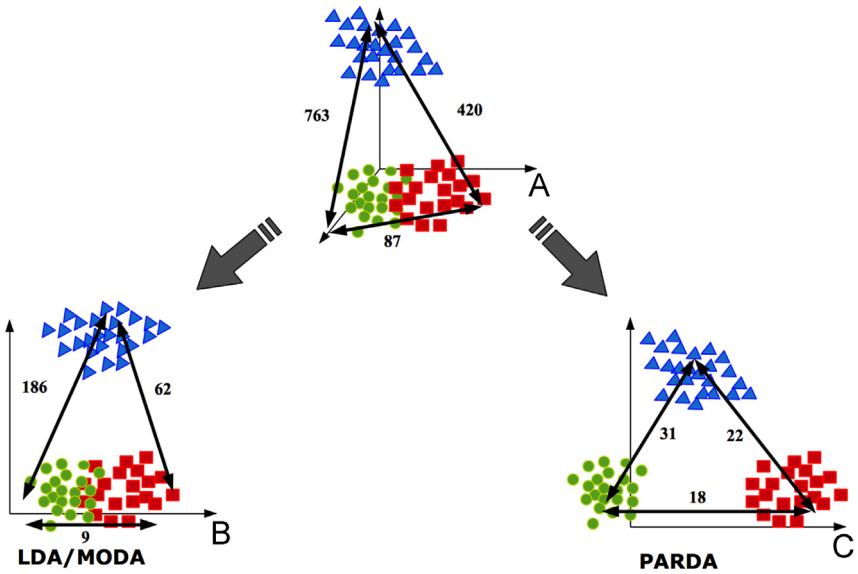

Fig. 1. (A) A Synthetic example of a 3-class problem with three dimensional data; L_1 triangles, L_2 squares, and L_3 circles. The numbers shown on arrows indicate the KL divergence between classes. The contribution of each pairwise divergence to the total divergence is 60%, 33%, and 7% for (L_1,L_2), (L_1,L_3), and (L_2,L_3), respectively. (B) and (C) Projections using MODA and PARDA, respectively, on two-dimensional subspaces. Note that the divergences in the lower dimensional subspaces are always less than the divergences in the original input space. This is due to the information loss incurred from the linear transformation and it shall be explained in Section 4.3. For MODA's projection, the contribution of each pairwise divergence to the total divergence is 72%, 28%, and 3% for the same class ordering. Note that the largest KL divergence in the input space increased in the lower dimensional subspace, and that the other (less) KL divergences became even smaller in the lower dimensional subspace – which is the class masking effect. For PARDA's projection, the contribution of each pairwise divergence to the total divergence is 44%, 31%, and 25% for the same class ordering. Note that, while MODA decreases the separation from 7% to 3% for (L_2,L_3), PARDA increases the separation to 25%.

domains, most notably face recognition (see [4,11–14] for a good review of these methods).

Of particular interest is the extension proposed by De La Torre and Kanade [15], namely Multimodal Oriented Discriminant Analysis (MODA), where it was shown that FDA's objective function is a special case of a more general objective that maximizes the Kullback–Leibler (KL) divergence [16] between two Gaussian densities, when the two Gaussians share the same covariance matrix. Note that the symmetric KL divergence considers the difference in mean locations and the difference in covariance matrices (size and orientation). Therefore, MODA searches for a linear transformation that maximizes the symmetric KL divergence between the two classes in the low dimensional subspace.

To account for the multiclass heteroscedastic case, MODA sums over all KL divergences between every pair of different classes and maximizes that sum in the lower dimensional subspace. This is similar to LDA's objective function which, as shown by Loog et al. [11], maximizes the sum of pairwise FDAs between all pairs of different classes. Hence MODA is a consistent generalization of FDA/LDA to multimodal Gaussian distributions with different means and covariance matrices.

However, as noted by several researchers [11,12,17,18], even if all the homoscedastic assumptions are satisfied, LDA and MODA suffer from the serious problem of merging classes that are close to each other in the original input space, *a.k.a* the class masking problem. This is due to the fact that LDA and MODA shift the 2-class problem to the multiclass setting by maximizing the sum of all KL divergences, which is a suitable objective function when all classes are equally distant from each other in terms of KL divergence.

Fig. 1A depicts a synthetic example for a 3-class problem with three dimensional data. Traditional methods like LDA or MODA find projections that maximize the sum of pairwise Mahalanobis distance (for LDA) or the KL divergence (for MODA) between pairwise classes. Note that the first term in the symmetric KL divergence – for two multivariate Gaussians see Eq. (6) – and the Mahalanobis distance (a special case from the KL divergence) are positive quadratic distance functions. From the optimization of minimax functions [19], it is known that the sum of positive powered functions, $\sum_{j=1}^{m}[f_j]^p$, where $p>1$, is a smooth approximation for $\max_{1\le j\le m}[f_j]^p$, as p is increasing, and hence $\sum_{j=1}^{m}[f_j]^p \approx [f_r]^p$ where $f_r > f_j \ \forall j \neq r$. Using this argument,[1] and for $p=2$, we argue that LDA is in fact maximizing a smooth approximation of the maximum of quadratic distances. Similarly, due to the quadratic distance in the first term of the symmetric KL divergence (in the case of Gaussians), MODA also maximizes a smooth approximation of the maximum divergences between Gaussians. Hence, LDA and MODA intrinsically prefer solutions that encourage maximizing the largest distance in the input space to make it even larger in the lower dimensional subspace, i.e., LDA and MODA put needless effort to maximize already distant classes in the input space. This effect can be seen in Fig. 1B, where MODA's projection gives relatively better increase in terms of KL divergence to the classes that are farther away in the input space, while it only makes a slight effort to separate between classes that are closer to each other in the input space.

1.1. Contribution

We note that the multiclass problem for LDA and MODA defines an independent objective function for each pair of different classes that needs to be optimized, namely maximize the symmetric KL divergence between every pair of different classes. Hence, the set of all pairs of different classes defines an optimization problem with *multiple objective functions* that share one final solution, and if possible, they all need to be *simultaneously optimized*. Given this perspective, maximizing the sum over all pairwise KL divergences (or quadratic distances) does not consider each objective function independently, since as explained above, maximizing that sum approximates a max function that only encourages maximizing the largest KL divergence. In other words, upgrading the problem of learning a discriminant subspace from the 2-class setting to the

[1] This will be explained in more detail in Section 4.

multiclass setting by summing over all pairwise KL divergences as in LDA/MODA is not the appropriate path to handle a multi-objective optimization problem.

Our contribution in this research stems from the above observation. In particular, we propose four models for multiclass heteroscedastic linear discriminant analysis (HDA) based on the theory of multiobjective optimization (MOP) [20–22]. Due to their parametrization, these objective functions can adapt to the class configuration[2] for any classification problem. While LDA and MODA's objectives pull apart the two classes with the largest KL divergence, PARDA, or Pareto Discriminant Analysis, encourage solutions in which all classes are equally spread from each other.

PARDA concentrates its effort on overlapping classes while it safeguards well separated classes from overlapping in the lower dimensional subspace. That is, PARDA puts more effort in maximizing the distance between classes that are closer in the projected space, and will relax the distance between classes that are farther away. Fig. 1C shows the projection obtained by PARDA in a two dimensional space. Unlike MODA, the 2D projection obtained by PARDA encourages the case where in the lower dimensional subspace, the class means are maximally separated from each other, and hopefully equally distant from each other as well, while the class overlap (due to class spread) is minimized.

Our paper is organized as follows. Following the introduction, Section 2 briefly covers two aspects in the LDA literature: LDA extensions to HDA, and the class merging (masking) problem. In order to make our paper self-contained, Section 3 covers all the necessary material the reader will need for multiobjective optimization. Section 4 introduces the model proposed in this paper, Pareto discriminant analysis (PARDA), and Section 5 extends PARDA to the case when the class distribution is non-Gaussian and multimodal. Experimental results are reported in Section 6, and concluding remarks with future research directions are drawn in Section 7.

2. Literature review

The literature on discriminant analysis (DA) is immense and a thorough review will be beyond the scope of the paper. We first review the basic notations for LDA, then focus on two research directions for DA; heteroscedastic and multiclass extensions of LDA, and the class masking problem.

2.1. LDA notations

We are given a data set $\mathcal{D}=\{(\mathbf{x}_i,\ell_i)_{i=1}^n\subseteq\mathbb{R}^d\times\mathcal{L}\}$ with labels $\ell_i\in\mathcal{L}=\{L_1,\ldots,L_c\}$ (find our notations' explanation in the footnote[3]). LDA's objective is to find a linear transformation matrix $\mathbf{B}\in\mathbb{R}^{d\times d_0}$, with $d_0\ll d$ such that the class means are maximally separated from each other, while the average spread of classes is minimized. Throughout the text, we will use the notion of *well separated classes* to imply that the class means are maximally separated from each other, while the class overlap due to class spread is minimized.

There are various objective functions that define the transformation matrix $\mathbf{B}$, from which Rayleigh type quotients are among the most popular LDA objective functions [2]. Some of these objective functions include

$$E_1(\mathbf{B})=\mathrm{tr}\left\{(\mathbf{B}^\top\mathbf{S}_2\mathbf{B})^{-1}(\mathbf{B}^\top\mathbf{S}_1\mathbf{B})\right\},\quad E_2(\mathbf{B})=\frac{\mathrm{tr}\{\mathbf{B}^\top\mathbf{S}_1\mathbf{B}\}}{\mathrm{tr}\{\mathbf{B}^\top\mathbf{S}_2\mathbf{B}\}},$$

$$E_3(\mathbf{B})=\frac{|\mathbf{B}^\top\mathbf{S}_1\mathbf{B}|}{|\mathbf{B}^\top\mathbf{S}_2\mathbf{B}|},$$

where matrix $\mathbf{S}_1$ can be any of the matrices $\{\mathbf{S}_b,\mathbf{S}_b,\mathbf{S}_t\}$, matrix $\mathbf{S}_2$ can be any of the matrices $\{\mathbf{S}_w,\mathbf{S}_t,\mathbf{S}_w\}$, and $\mathbf{S}_b$, $\mathbf{S}_w$ and $\mathbf{S}_t$ are known as the between-class scatter matrix, the within-class scatter matrix and the total-class scatter matrix, respectively. Note that $\mathbf{S}_b$ is a measure for the average distance between the means of all classes, while $\mathbf{S}_w$ is an average covariance matrix that acts as a measure of compactness for all classes. The upper bound on the ranks of $\mathbf{S}_b$, $\mathbf{S}_w$ and $\mathbf{S}_t$ is $\min(c-1,d)$, $\min(n-c,d)$ and $\min(n-1,d)$, respectively.

The objective function $E_1(\mathbf{B})$ is among the most popular LDA objective functions in the literature. However, this formulation is restricted to the original homoscedastic setting, i.e., the samples in each class L_j are assumed to have a Gaussian distribution $\mathcal{N}(\cdot;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$ with $\boldsymbol{\Sigma}_1=\cdots=\boldsymbol{\Sigma}_c=\boldsymbol{\Sigma}$. In practice, when the homoscedastic assumption does not hold, or when the classes are not Gaussians, all parameters are approximated by their sample estimates, and $\boldsymbol{\Sigma}$ is approximated by $\mathbf{S}_w$. Unfortunately, this approximation does not fully exploit the rich information in the heteroscedastic setting which lies in the covariance matrix of each class.

2.2. Heteroscedastic multiclass extensions of LDA

Campbell [5] was the first to develop a general formulation for LDA as a maximum likelihood (ML) estimation of the parameters of a Gaussian model. His model's structure relied on two assumptions: (i) all class means (or all discriminatory information between the classes) lie in a $(c-1)$-dimensional subspace of the original d-dimensional input (or feature) space; and (ii) all classes have equal covariance matrices (homoscedastic setting). Kumar and Andreou [7] extended Campbell's ML model to the heteroscedastic setting and named it HDA. Their objective function is the log-likelihood of the Gaussian models in the projected low dimensional subspace. By taking the gradient of this objective, they derive ML estimators for the class means and covariances in the low dimensional subspace. Hastie and Tibshirani [23] tried to work around the homoscedastic assumption of Campbell and proposed that each class can be modelled as mixtures of Gaussians while maintaining that all classes and sub-classes share a pooled covariance matrix. In a similar vein, Zhu and Martinze [13] proposed subclass discriminant analysis (SDA), in which the between class scatter matrix $\mathbf{S}_b$ is replaced by the *between subclass* scatter matrix, where each class now is divided (by means of a clustering algorithm) into several subclasses. In addition, the authors propose two criteria to select the number of subclasses that maximizes the classification accuracy.

Tou and Heydorn derived LDA's objective function $E_1(\mathbf{B})$ in Section 2.1 from maximizing the symmetric KL divergence between two Gaussian densities under the heteroscedastic assumption. Independently, De La Torre and Kanade [15] proposed MODA which is a more general formulation than the one proposed in [24] since it considers a multiclass heteroscedastic setting, and that each class is a mixtures of Gaussians (i.e. multimodal).

Saon et al. [25] maximize an objective function based determinants' ratio:

$$E_{\mathrm{SAON}}(\mathbf{B})=\prod_{j=1}^{c}\left(\frac{|\mathbf{B}^\top\mathbf{S}_B\mathbf{B}|}{|\mathbf{B}^\top\boldsymbol{\Sigma}_j\mathbf{B}|}\right)^{n_j},\tag{1}$$

which is a weighted product of each individual direction (or dimension in the low dimensional subspace) of the data. This objective function models the data orientation (or directionality) and has the property of being invariant to transformations to the

[2] The relative location of classes to each other in the input space.

[3] Notations: Bold capital letters denote matrices ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$). Bold lower-case letters denote column vectors ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$). Non-bold lower case letters represent scalar variables (a,b,c) or indexes (i,j,k). Sets are denoted by calligraphic upper case letters ($\mathcal{X}$, $\mathcal{Y}$, $\mathcal{R}$). Spaces are denoted by double–bold upper case letters ($\mathbb{R}$, $\mathbb{S}$). $\mathbf{I}$ is the identity matrix of suitable dimension. $tr(\mathbf{A})=\sum_i a_{ii}$ is the trace of the matrix $\mathbf{A}$, and $|\mathbf{A}|$ is the determinant of matrix $\mathbf{A}$. The multivariate Gaussian distribution is denoted by $\mathcal{N}(\cdot;\boldsymbol{\mu},\boldsymbol{\Sigma})$ with mean vector $\boldsymbol{\mu}\in\mathbb{R}^d$, covariance matrix $\boldsymbol{\Sigma}\in\mathbb{S}_{++}^{d\times d}$, and $\mathbb{S}_{++}^{d\times d}$ is the space of symmetric and positive definite (PD) matrices.

range of the solution (eigenvectors). In addition, similar to LDA and HDA, it is invariant to linear transformations of the data in the input space. Note that each column of $\mathbf{B}$ in E_{SAON} corresponds to one class. Zhu and Hastie [8] proposed a feature extraction criterion for nonparametric DA, which generalizes the Fisher criterion when the data for each class is not Gaussian. For a fixed direction $\mathbf{b}$, they define the marginal generalized log-likelihood-ratio (LR) statistic:

$$LR(\mathbf{b}) = \log \frac{\max_{\Pr_{L_j}} \prod_{j=1}^{c} \prod_{\mathbf{x}_i \in L_j} \Pr_{L_j}^{(\mathbf{b})}(\mathbf{b}^\top \mathbf{x}_i)}{\max_{\Pr_{L_j} = \Pr} \prod_{j=1}^{c} \prod_{\mathbf{x}_i \in L_j} \Pr^{(\mathbf{b})}(\mathbf{b}^\top \mathbf{x}_i)}, \tag{2}$$

where $\Pr_{L_j}^{(\mathbf{b})}(\cdot)$ is the marginal density along the projection defined by $\mathbf{b}$ for class L_j, and $\Pr^{(\mathbf{b})}(\cdot)$ is the corresponding marginal density under the null hypothesis that classes share the same density function. Note that if for all classes L_j, $\Pr_{L_j}^{(\mathbf{b})}(\mathbf{x}) \sim \mathcal{N}_j(\boldsymbol{\mu}_j, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_j$ for $1 \le j \le c$, then the discriminant directions maximizing $LR(\mathbf{b})$ are equivalent to Fisher's LDA (see Result 1 in [8]).

Loog and Duin [26] consider the multiclass heteroscedastic setting using the Chernoff distance which is a symmetric divergence measure between probability distributions. The Chernoff distance, similar to the symmetric KL divergence employed in this work, considers the means and covariance matrices when measuring the discrimination between classes. The authors formulate the Chernoff distance between two classes (modelled as Gaussians) as a trace function of a symmetric PD matrix – the directed distance matrix (DDM) – which yields more than one direction (its eigenvectors) that can discriminate between the two different Gaussians. For the multiclass case, they use the DDM as a building block in the multiclass weighted pairwise formulation of [11] (discussed below), and the final linear transformation $\mathbf{B}$ is obtained by means of an eigenvalue problem for the final DDM.

Recently, Hamsici and Martinez [4] find a Bayes optimal discriminant direction for multiclass ($c > 2$) homoscedastic problems. Their solution is based on realizing that the class projected whitened means on a single discriminant direction $\mathbf{b}$ have the same ordering for a range of other directions $\mathbf{b}'$. This set of discriminant directions defines a convex polyhedron on which the Bayes error function is convex as well, and hence can be minimized by standard convex optimization algorithms. For the multiclass heteroscedastic Gaussian case, they extend their result using the kernel trick, and to obtain a d_0-dimensional subspace, their approach is repeated recursively on the null space of the previous projection directions. Gao et al. [27], motivated by graph embedding approaches and manifold learning algorithms [28], propose an enhanced Fisher discriminant criterion (EFDC) based on modelling the within class variability by means of neighbourhood graphs. EFDC find bases matrix $\mathbf{B}$ that maximizes

$$\left(\mathbf{B}^\top \mathbf{S}_w \mathbf{B}\right)^{-1} \left(\mathbf{B}^\top [\alpha \mathbf{S}_b + (1-\alpha)\mathbf{S}_d] \mathbf{B}\right), \tag{3}$$

where $\mathbf{S}_d = \mathbf{X}^\top (\mathbf{D} - \mathbf{K})\mathbf{X}$, $\mathbf{K}$ is a kernel (and/or adjacency) matrix that encodes the similarity using Gaussian kernels between points in the same class, $\mathbf{D} = \text{diag}(\mathbf{K}\mathbf{1}_n)$, $\mathbf{X}_{n \times d}$ is the data matrix, and $0 < \alpha < 1$ is a tuning parameter that controls the balance between the discriminative information in $\mathbf{S}_b$ and the similarity information in $\mathbf{S}_d$. Recall that $\mathbf{S}_b$ is based on the heteroscedastic Gaussian assumption, while $\mathbf{S}_d$ can be seen as carrying density information based on a kernel density estimate for the data, which is a nonparametric approach. Hence $\mathbf{S}_b$ and $\mathbf{S}_d$ are complimentary to each other since $\mathbf{S}_d$ together with α relax the Gaussian assumption in $\mathbf{S}_b$, thereby increasing the stability of LDA when dealing with real world data sets.

2.3. The class masking problem

To solve the class masking problem, Lotlikar and Kothari [29] proposed fractional step DA (F-LDA) where the dimensionality is reduced in fractional steps, i.e., iteratively from d to $d-1$ (one dimension at a time) while applying proper weighting on the data in order to avoid the class masking problem. Lu and Plataniotis [12], in a two-stage algorithm, proposed a weighted variant of direct-LDA [30] combined with fractional step LDA [29]. For the between-class scatter matrix $\mathbf{S}_b$, they applied weights that are inversely proportional to the distance between class means. Alternatively, Loog et al. [11] suggested that the weights applied to $\mathbf{S}_b$ should link the distance between the class means to the amount of error they cause. Therefore, the weight between two classes is measured as $(1/2\delta_{ij})\text{erf}(\delta_{ij}/2\sqrt{2})$, where $\text{erf}(\cdot)$ is the error function and δ_{ij} is the Euclidean distance between class means i and j in the whitened space. In Section 4.9, we will discuss other approaches for the class masking problem [31,32] and see how the proposed Pareto model differs from these approaches.

3. Multiobjective optimization

Multiobjective optimization (MOP), or vector optimization (VOP), is a branch of optimization science that is concerned with the simultaneous optimization of more than one objective function. In real world applications, it is often the case that the objectives are contradictory in a way that optimizing one of the objective functions entails the inefficiency of another one. In such cases, one would require a good compromise solution which is suboptimal but acceptable as much as possible to the individual objective functions. MOP is the science that can find this good compromise solution [21].

Let $\mathbf{f}(\boldsymbol{\theta}) = [f_1(\boldsymbol{\theta}) \ldots f_\kappa(\boldsymbol{\theta})]^\top$ be the vector valued objective function to be optimized where $\mathbf{f}(\boldsymbol{\theta}) \in \mathbb{R}^\kappa$, $\boldsymbol{\theta} \in \mathcal{R} \subseteq \mathbb{R}^p$ is the parameter vector for the set of objective functions, $f_j(\boldsymbol{\theta}) \in \mathbb{R}$ is the jth objective function, $\mathcal{R}$ is the feasible set for the values of the parameter vector $\boldsymbol{\theta}$, and $\mathbb{R}^\kappa$ is the objective space. For the sake of a consistent discussion in this section, we will consider that our objective is to minimize[4] $\mathbf{f}(\boldsymbol{\theta})$. The goal of VOP is to find $\boldsymbol{\theta}^*$ that simultaneously minimizes all $f_j(.)$'s. In practice, the individual objective functions can be in contradiction to each other, i.e., an improvement with regard to one objective can cause the deterioration of at least another objective function. For this, a formal definition is needed for the task of VOP, the order relation "$\le$" in VOP, *efficient points*, and *Pareto optimal points*.

Definition (*Order relation "$\le$" in the objective space* $\mathbb{R}^\kappa$). Let $\mathbf{z}_1$ and $\mathbf{z}_2$ be two points in the objective space $\mathbb{R}^\kappa$. The order relation "$\le$" is defined as $\mathbf{z}_1 \le \mathbf{z}_2 \Longleftrightarrow \mathbf{z}_2 - \mathbf{z}_1 \in \mathbb{R}^\kappa_+$, where $\mathbb{R}^\kappa_+ = \{\mathbf{z} \in \mathbb{R}^\kappa | z_i \ge 0, \text{ and } 1 \le i \le \kappa\}$ is the nonnegative orthant of $\mathbb{R}^\kappa$ and, $\forall i \in \{1, \ldots, \kappa\}$, $z_1^i \le z_2^i$, $\exists j \in \{1, \ldots, \kappa\}$ s.t. $z_1^j < z_2^j$.

Definition (*Efficient point*). Let $\mathcal{Z} = \mathbf{f}(\mathcal{R}) \subseteq \mathbb{R}^\kappa$ be the image of the feasible set $\mathcal{R} \subseteq \mathbb{R}^d$ in the objective space. A point $\mathbf{z}^* \in \mathcal{Z}$ is called an *efficient point* with regards to the order relation "$\le$" defined above on $\mathbb{R}^\kappa$, iff there exists no other $\mathbf{z} \in \mathcal{Z}$ s.t. $\mathbf{z} \le \mathbf{z}^*$ and $\mathbf{z} \ne \mathbf{z}^*$.

Definition (*Pareto optimal point*). A feasible point $\boldsymbol{\theta}^* \in \mathcal{R}$, where $\mathcal{R}$ is the feasible set for $\boldsymbol{\theta}$, is called *Pareto optimal* iff $\mathbf{z}^* = \mathbf{f}(\boldsymbol{\theta}^*)$ is an *efficient point*.

VOP is formally defined as finding *efficient points* $\mathbf{z}^* \in \mathcal{Z}$ with regard to the order relation "$\le$" on $\mathbb{R}^\kappa$, along with their *Pareto optimal points* $\boldsymbol{\theta}^*$ pertaining to them [22]. It could be the case that all objective functions are of equal importance. In this case, the best that VOP can do is to provide the decision maker a set of all *efficient points* along with their *Pareto optimal points* pertaining to them. These two sets are known as the *Efficient Set* and the *Pareto Front* (or *Pareto Set*), respectively.

[4] Inverting the discussion on maximizing $\mathbf{f}(\boldsymbol{\theta})$ can be simply done by minimizing $-\mathbf{f}(\boldsymbol{\theta})$.

There are various techniques for solving VOP problems [21,22], and a class of these techniques form what are known as deterministic methods. These methods scalarize the vector optimization problem through a parametric formulation and then solve the new objective function using standard optimization techniques. From the deterministic methods, the weighted-sum method and the weighted L_p-Metric method were found to be well studied with theoretical results that guarantee *Pareto optimal solutions*.

3.1. The weighted-sum method

The weighted-sum (WS) method assigns a weight w_j to each objective function such that $w_j \geq 0$, $1 \leq j \leq \kappa$, and $\sum_{j=1}^{\kappa} w_j = 1$; i.e. a convex combination of the objective functions. The final optimization problem to be solved is

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \mathcal{R}}{\operatorname{argmin}} \mathbf{w}^\top \mathbf{f}(\boldsymbol{\theta}), \tag{4}$$

where $\mathbf{w} = [w_1 \ldots w_\kappa]^\top$. The weight w_j reflects the significance of the individual objective function $f_j(\cdot)$, and hence, it can reflect some *a priori* knowledge from the problem domain or, impose some bias on the final solution $\boldsymbol{\theta}^*$. By varying the weight vector $\mathbf{w}$, one can obtain a subset of the *efficient set* and its pertaining subset of *Pareto optimal solutions*. We state here Theorem (4.1) from [22] that guarantees a *Pareto optimal solution* for the WS method for Problem (4).

Theorem 3.1. *Let $\boldsymbol{\theta}^* \in \mathcal{R}$ be an optimal solution of (4), then the following statements hold: (i) If $\mathbf{w} \geq 0$, then $\boldsymbol{\theta}^*$ is a Pareto optimal point. (ii) If $\mathbf{w} \geq 0$ and $\boldsymbol{\theta}^*$ is a unique optimal solution of (4), then $\boldsymbol{\theta}^*$ is a global Pareto optimal point. (iii) If $\mathbf{w} > 0$, i.e., all its components are strictly greater than zero, then $\boldsymbol{\theta}^*$ is a proper Pareto optimal point. (iv) If $\mathbf{w} > 0$ and $\boldsymbol{\theta}^*$ is a unique optimal solution of (4), then $\boldsymbol{\theta}^*$ is a strong Pareto optimal point.*[5]

The WS method, however, has an implicit assumption which can easily be a drawback in practice. The method requires that $\mathcal{Z} = \mathbf{f}(\mathcal{R})$ be a convex set. In practice the set $\mathcal{Z}$ might not be convex and as a side effect, there will be a set of efficient solutions $\mathbf{z}^*$ that cannot be found using the WS method.

3.2. The L_p-metric method

In an ideal situation, the objective of VOP is to achieve the optimal solution for each individual objective function $f_j(\cdot)$. Let $\mathbf{t}^* \in \mathbb{R}^\kappa$ be such an ideal target point in the objective space. Then, $\forall \mathbf{z} \in \mathcal{Z}$, $\mathbf{t}^* \leq \mathbf{z}$ and $\mathbf{t}^*$ might or might not be in $\mathcal{Z}$. Since in real world problems, the individual objectives might conflict with each other, achieving $\mathbf{t}^*$ is usually impossible, however it can serve as a reference point with the goal of seeking a solution as close as possible to $\mathbf{t}^*$ (see Fig. 2). Note that t^* is also known as the *Utopia* point. Formally, given a distance function $dist : \mathbb{R}^\kappa \times \mathbb{R}^\kappa \to \mathbb{R}_+$, the L_p-Metric method is given by $\min_{\boldsymbol{\theta} \in \mathcal{R}} dist(\mathbf{f}(\boldsymbol{\theta}) - \mathbf{t}^*)$. Since the objective space $\mathbb{R}^\kappa$ is endowed with a vector norm $\|\cdot\|$ then the induced weighted distance, or the L_p–Metric method can be defined as follows:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \mathcal{R}}{\operatorname{argmin}} d(\boldsymbol{\theta}) \quad \text{where}$$

$$d(\boldsymbol{\theta}) = \left(\sum_{j=1}^{\kappa} w_j |f_j(\boldsymbol{\theta}) - t_j^*|^p \right)^{1/p}, \tag{5}$$

$p \in [1, \infty]$, $w_j > 0$ is the weight for the j-th objective function, and $\sum_{j=1}^{k} w_j = 1$. Similar to the WS method, the weight w_j reflects the significance of the objective function $f_j(\cdot)$. Note that the WS method can be considered as a special case from the L_p-Metric method. Also note that by definition of the L_p-Metric method, the weight vector $\mathbf{w} > 0$, and according to (3.1), $\boldsymbol{\theta}^*$ will be at least a *properly Pareto optimal* solution. We now state Theorem (4.20) from [22] (see proof in p. 112) that links the monotonicity of a norm to the solution obtained by Problem (5) in order to introduce the main result in Corollary 3.3.

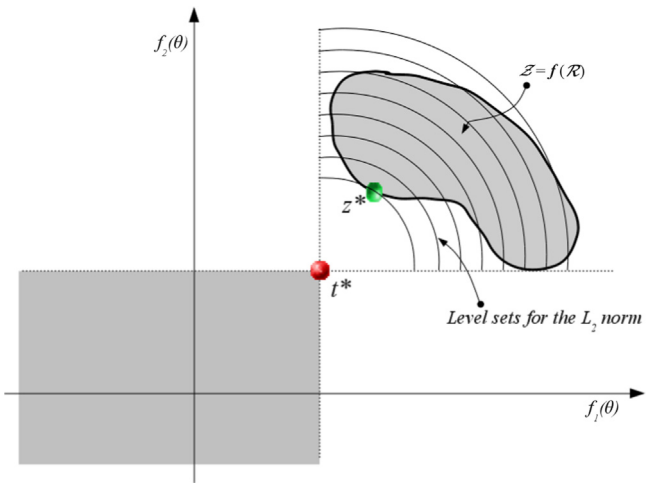

Fig. 2. The intersection of level sets for the L_p-Metric method for $p=2$ and with $\mathcal{Z} = \mathbf{f}(\mathcal{R})$ in the objective space. Note that the ideal point $\mathbf{t}^* \notin \mathbf{f}(\mathcal{R})$ and the efficient point $\mathbf{z}^*$ is the closest to it.

Theorem 3.2 (*4.20 in Ehrgott [22]*). *If $\|\cdot\|$ is a strictly monotonic norm and $\boldsymbol{\theta}^*$ is an optimal solution of Problem (5), then $\boldsymbol{\theta}^*$ is Pareto optimal.*

Corollary 3.3. *For the L_p norm $\|\cdot\|_p$, if $1 \leq p < \infty$ and $\boldsymbol{\theta}^*$ is the optimal solution for Problem (5), then $\|\cdot\|_p$ is strictly monotonic and $\boldsymbol{\theta}^*$ is Pareto optimal.*

The L_p-Metric method has a nice interpretation in terms of level sets $\{\mathbf{z} \in \mathbb{R}^\kappa | \|\mathbf{z} - \mathbf{t}^*\|_p \leq u\}$ where such sets contain all points of distance u or less to t^*. From this perspective, the goal of the L_p-Metric method, illustrated in Fig. 2, is to search for the smallest u such that the intersection of the corresponding level set with $\mathcal{Z} = \mathbf{f}(\mathcal{R})$ is nonempty.

The two methods presented so far will be the models that encapsulate the multiclass HDA problem. In the following, we derive a multiclass HDA formulation that will fit in these two models, and then propose PARDA in Section 4.2.

4. Pareto discriminant analysis

We begin our discussion with a formulation for multiclass HDA that generalizes the LDA formulation presented in Section 2.1. This formulation will allow a clear understanding for the class masking problem, and will naturally fit in the MOP models presented in the previous section.

For two classes, L_i and L_j, each modelled as a Gaussian distribution $\mathcal{N}_i$ and $\mathcal{N}_j$, respectively, the separability or discriminability between the two classes can be measured using the symmetric KL divergence defined as

$$\begin{aligned} J_{KL}(\mathcal{N}_i, \mathcal{N}_j) &= \frac{1}{2} \int \left(\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) - \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \right) \log \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} d\mathbf{x}, \\ &= \frac{1}{2} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^\top (\boldsymbol{\Sigma}_i^{-1} + \boldsymbol{\Sigma}_j^{-1})(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2} \operatorname{tr}\left(\boldsymbol{\Sigma}_i \boldsymbol{\Sigma}_j^{-1} + \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\Sigma}_j - 2\mathbf{I} \right). \end{aligned} \tag{6}$$

[5] We will rely on this property of the weight vector when this method will be presented in the context of LDA in Section 4.

Note that (6) measures the difference between two Gaussians in terms of the differences in means (positions), and covariances (size and orientation). The divergence, in general, is a measure of distance or separability between probability distributions, and hence, the larger the interclass divergence $J_{KL}(\mathcal{N}_i,\mathcal{N}_j)$, the greater the separation between $\mathcal{N}_i$ and $\mathcal{N}_j$.

Let $\mathbf{u}_{ij}=\boldsymbol{\mu}_i-\boldsymbol{\mu}_j$ and $\mathbf{U}_{ij}=\mathbf{u}_{ij}\mathbf{u}_{ij}^\top$, then (6) can be rewritten as

$$J_{KL}(\mathcal{N}_i,\mathcal{N}_j)=\frac{1}{2}\mathrm{tr}\{\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\Sigma}_j+\mathbf{U}_{ij})\}+\frac{1}{2}\mathrm{tr}\left\{\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{\Sigma}_i+\mathbf{U}_{ij})\right\}-d. \tag{7}$$

For the c-class problem, the total divergence will be

$$\begin{aligned}E_{Tot}&=\sum_{i=1}^{c-1}\sum_{j=i+1}^{c}J_{KL}(\mathcal{N}_i,\mathcal{N}_j),\\&=\frac{1}{2}\mathrm{tr}\left\{\sum_{i=1}^{c}\boldsymbol{\Sigma}_i^{-1}\mathbf{S}_i\right\}-\frac{c(c-1)}{2}d,\end{aligned} \tag{8}$$

where

$$\mathbf{S}_i=\sum_{j=1\,j\neq i}^{c}[\boldsymbol{\Sigma}_j+\mathbf{U}_{ij}]. \tag{9}$$

MODA seeks a linear transformation $\mathbf{B}\in\mathbb{R}^{d\times d_0}$ with $d_0\ll d$ such that E_{Tot} in the lower dimensional subspace is maximized. Note that the columns of $\mathbf{B}$ form the bases for the sought low dimensional subspace, and that $\mathbf{B}$ can have any number of bases d_0 such that $1\leq d_0\leq d-1$. This is unlike FDA/LDA that can only define subspaces of dimensionality $d_0\leq\min(c-1,d-1)$. In the lower dimensional subspace, classes L_i and L_j will be projected as $\mathcal{N}_i(\mathbf{B}^\top\boldsymbol{\mu}_i,\mathbf{B}^\top\boldsymbol{\Sigma}_i\mathbf{B})$ and $\mathcal{N}_j(\mathbf{B}^\top\boldsymbol{\mu}_j,\mathbf{B}^\top\boldsymbol{\Sigma}_j\mathbf{B})$, respectively, and hence

$$\begin{aligned}J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B})=&\frac{1}{2}\mathrm{tr}\left\{(\mathbf{B}^\top\boldsymbol{\Sigma}_i\mathbf{B})^{-1}\mathbf{B}^\top(\boldsymbol{\Sigma}_j+\mathbf{U}_{ij})\mathbf{B}\right\}\\&+\frac{1}{2}\mathrm{tr}\left\{(\mathbf{B}^\top\boldsymbol{\Sigma}_j\mathbf{B})^{-1}\mathbf{B}^\top(\boldsymbol{\Sigma}_i+\mathbf{U}_{ij})\mathbf{B}\right\}-d_0,\end{aligned} \tag{10}$$

and for the c-class problem, the total divergence will be

$$\begin{aligned}E_{MODA}(\mathbf{B})&=\sum_{i=1}^{c-1}\sum_{j=i+1}^{c}J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}),\\&=\frac{1}{2}\mathrm{tr}\left\{\sum_{i=1}^{c}(\mathbf{B}^\top\Sigma_i\mathbf{B})^{-1}(\mathbf{B}^\top\mathbf{S}_i\mathbf{B})\right\}-\frac{c(c-1)}{2}d_0,\end{aligned} \tag{11}$$

which defines MODA's objective function. The optimal $\mathbf{B}^*_{MODA}$ is then

$$\mathbf{B}^*_{MODA}=\underset{\mathbf{B}}{\mathrm{argmax}}\;E_{MODA}(\mathbf{B}), \tag{12}$$

which is optimized using a gradient ascent procedure. Note that maximizing Problem (12) under the assumption that $\boldsymbol{\Sigma}_i=\boldsymbol{\Sigma}_j=\boldsymbol{\Sigma}$, for $1\leq i,j\leq c$, yields the standard LDA formulation (up to a scaling factor) for the multiclass problem. Based on this general formulation for the multiclass HDA, we can proceed to the class masking problem and see how it becomes a natural manifestation from such a formulation.

4.1. The class masking problem

In the 2-class setting, LDA (i.e. $\boldsymbol{\Sigma}_1=\boldsymbol{\Sigma}_2$) and MODA (i.e. $\boldsymbol{\Sigma}_1\neq\boldsymbol{\Sigma}_2$) search for an optimal $\mathbf{B}^*$ that maximizes $J_{KL}(\mathcal{N}_1,\mathcal{N}_2;\mathbf{B})$ in (10). To account for the multiclass setting, LDA/MODA use the same objective function that sums over all pairwise KL divergences and searches for $\mathbf{B}^*$ that maximizes the total divergence $E_{MODA}(\mathbf{B})$ in (11).

Note that the original KL divergence in (6) has in fact two terms: (i) the first term, which is a quadratic distance, measures the difference between the means $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ weighted by the covariance matrices, and (ii) the second term, which is independent of $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$, only measures the discrepancy (or dissimilarity) between $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_j$ [16]. If $\boldsymbol{\Sigma}_i=\boldsymbol{\Sigma}_j$, the second term will be zero and (6) reduces to a quadratic distance. Hence, the second term increases the final $J_{KL}(\mathcal{N}_1,\mathcal{N}_2;\mathbf{B})$ whenever there is a disagreement between $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_j$.

From the optimization of minimax functions [19], it is known that the sum of positive powered smooth positive functions, $\sum_{j=1}^{m}[f_j]^p$, where $p>1$, is a smooth approximation for $\max_{1\leq j\leq m}[f_j]^p$, as p is increasing. That is, for large $p\geq 2$, $\sum_{j=1}^{m}[f_j]^p\approx[f_r]^p$ where $f_r>f_j\ \forall j\neq r$.

Using this argument, let $f=\|\boldsymbol{\mu}_i-\boldsymbol{\mu}_j\|^p_{\mathbf{A}}$. Then for $p=2$ and $\mathbf{A}=(\boldsymbol{\Sigma}_i^{-1}+\boldsymbol{\Sigma}_j^{-1})$, we get $(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\top\mathbf{A}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)$ which is the quadratic distance in (6). It is possible to see that LDA (i.e. $\boldsymbol{\Sigma}_i=\boldsymbol{\Sigma}_j$) intrinsically prefers solutions that encourage maximizing the largest quadratic distance between $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$, to make it larger at the output space. Similarly, due the first and second terms in (6) as explained above, we argue that MODA's objective function is a smooth approximation for $E_{MODA}(\mathbf{B})\approx\max_{1\leq i,j\leq c}J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B})$.

Hence, MODA also encourages solutions that maximize the largest KL divergence to make it larger at the output space. Therefore, shifting the problem from the two-class setting to the multiclass setting using this scalarization technique (plain sum over all objective functions) intrinsically yields the class masking problem. Based on this understanding, it is possible to introduce the PARDA model and see how its mechanism counteracts the class masking effect.

4.2. A multiobjective optimization model for HDA

We propose different scalarization functions for the multiclass HDA problem using the MOP models introduced in the previous section. Since each pair of classes, $\mathcal{N}_i$ and $\mathcal{N}_j$, $1\leq i,j\leq c$, $i\neq j$, define their own individual objective function $J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B})$, then all $\kappa=c(c-1)/2$ pairs of classes result in κ objective functions that need to be simultaneously optimized. Since it is expected that the objective functions can conflict with each other, MOP models can guarantee that the obtained subspace will be in maximal agreement with all pairwise objective functions, but suboptimal for each individual objective function.

Using the WS method or the L_p-Metric method, and setting the appropriate weight vector for each model, the optimization effort will be distributed according to the relative position of classes to each other (with respect to class means together with covariance matrices). The simultaneous optimization of the objective functions will put more effort on overlapping classes while safeguards distant classes from overlapping in the lower dimensional subspace.

This is the major difference between MODA and LDA on one side and PARDA on the other side. While MODA (and LDA in the homoscedastic case) sum over all $J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B})$'s, and search for a bases that maximizes that sum, PARDA uses all the pairwise objective functions in a multiobjective optimization model and searches for a bases that is in maximal agreement with all pairwise objective functions and simultaneously maximizes them.

Formally, using the scalarization of the WS method in Problem (4), with (3.1), Pareto discriminant analysis can be defined using the following optimization problem:

$$\begin{aligned}&\mathbf{B}^*_{PDAWS}=\underset{\mathbf{B}\in\mathcal{R}}{\mathrm{argmax}}\;\mathcal{E}_{PDAWS}(\mathbf{B})\quad\text{where}\\&\mathcal{E}_{PDAWS}(\mathbf{B})=\sum_{i=1}^{c-1}\sum_{j=i+1}^{c}w_{ij}J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}),\\&\text{s.t.}\quad\sum_{i,j}w_{ij}=1,w_{ij}>0.\end{aligned} \tag{13}$$

Similarly, using the scalarization of the L_p-Metric method in (5), with Corollary 3.3, another Pareto discriminant analysis can be defined using the following optimization problem:

$$\mathbf{B}^*_{PDALP}=\underset{\mathbf{B}\in\mathcal{R}}{\mathrm{argmin}}\;\mathcal{L}_{PDALP}(\mathbf{B})\quad\text{where}$$

$$\mathcal{L}_{PDALP}(\mathbf{B}) = \sum_{i=1}^{c-1}\sum_{j=i+1}^{c} w_{ij}[J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}) - t^*_{ij}]^2,$$
$$s.t. \quad \sum_{i,j} w_{ij} = 1, w_{ij} > 0, \tag{14}$$

and $\mathcal{R} \subseteq \mathbb{R}^{d\times d_0}$. The subscript *PDA* stands for Pareto Discriminant Analysis, while *WS* and *LP* stand for the WS and L_p-Metric methods, respectively. According[6] to Theorem 3.1 and Corollary 3.3, and given a proper target vector $\mathbf{t}^*$, $\mathbf{B}^*_{PDAWS}$ and $\mathbf{B}^*_{PDALP}$ will be *Pareto Optimal*.

4.3. Information loss in DA

We consider now the information loss incurred from the linear transformation $\mathbf{y} = \mathbf{B}^\top \mathbf{x}$, for any $\mathbf{B} \in \mathbb{R}^{d\times d_0}$ with rank d_0, and how it relates $J_{KL}(\mathcal{N}_i,\mathcal{N}_j)$ to $J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B})$. Since the transformation $\mathbf{y} = \mathbf{B}^\top \mathbf{x}$ produces a linear combination of the components of the input vector $\mathbf{x}$, it can be shown that, in general, there is an information loss [16,33], and hence

$$J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}) \le J_{KL}(\mathcal{N}_i,\mathcal{N}_j), \quad \forall i,j, \tag{15}$$

and the amount of loss is

$$J_{KL}(\mathcal{N}_i,\mathcal{N}_j) - J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}) \ge 0. \tag{16}$$

Summing over all classes in (15), and recalling (8), (11), and (12) it turns that

$$E_{MODA}(\mathbf{B}^*_{MODA}) \le E_{Tot}. \tag{17}$$

Similarly, it is easy to verify that for $\mathbf{B}^*_{PDAWS}$ and $\mathbf{B}^*_{PDALP}$ the following holds:

$$J_{PDAWS} \equiv \sum_{i=1}^{c-1}\sum_{j=i+1}^{c} J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}^*_{PDAWS}) \le E_{Tot} \quad \text{and} \tag{18}$$

$$J_{PDALP} \equiv \sum_{i=1}^{c-1}\sum_{j=i+1}^{c} J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}^*_{PDALP}) \le E_{Tot}. \tag{19}$$

Two aspects are related to the discussion above. First, recall Figs. 1B and C and note that the divergences between different classes are smaller than their corresponding values in Fig. 1A. This is due to the impact of the linear dimensionality reduction (via matrix $\mathbf{B}$) on the divergence as expressed in inequality (15). Although the total divergence is reduced, note that due to the class masking effect, LDA/MODA have increased the separation between L_1 and L_2 from 60% to 72%, and reduced the separation between other classes.

Second, note that from the inequalities in (18) and (19), it is not possible to select the weights w_{ij} and the targets t_{ij} based on the KL divergences in the input space since these values will be very large compared to the divergences in the low dimensional subspace and hence, will mask all the differences between classes. These aspects will be explained in the following sections.

4.4. Solving PARDA

Unlike the closed form solution for $E_1(\mathbf{B})$ in (2.1) as a generalized eigenvalue problem (GEP), the objective functions for $\mathcal{E}_{PDAWS}(\mathbf{B})$ and $\mathcal{L}_{PDALP}(\mathbf{B})$ in (13) and (14), respectively, do not have such a feature. Further, $\mathcal{E}_{PDAWS}(\mathbf{B})$ and $\mathcal{L}_{PDALP}(\mathbf{B})$ are not convex in $\mathbf{B}$, and therefore, only a local optimum solution can be achieved. To solve PARDA, we use an iterative procedure based on gradient ascent (descent), with multiple restarts, since the gradient of $J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B})$ with respect to $\mathbf{B}$ has a closed form expression. For maximizing the energy $\mathcal{E}_{PDAWS}(\mathbf{B})$, the gradient step is

$$\mathbf{B}^{t+1} = \mathbf{B}^t + \eta_1 \frac{\partial}{\partial \mathbf{B}} \mathcal{E}_{PDAWS}(\mathbf{B}) \quad \text{where} \tag{20}$$

$$\frac{\partial}{\partial \mathbf{B}} \mathcal{E}_{PDAWS}(\mathbf{B}) = \sum_{i=1}^{c-1}\sum_{j=i+1}^{c} w_{ij} \frac{\partial}{\partial \mathbf{B}} J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}). \tag{21}$$

For minimizing the loss $\mathcal{L}_{PDALP}(\mathbf{B})$, the gradient step is

$$\mathbf{B}^{t+1} = \mathbf{B}^t - \eta_2 \frac{\partial}{\partial \mathbf{B}} \mathcal{L}_{PDALP}(\mathbf{B}) \quad \text{where} \tag{22}$$

$$\frac{\partial}{\partial \mathbf{B}} \mathcal{L}_{PDALP}(\mathbf{B}) = \sum_{i=1}^{c-1}\sum_{j=i+1}^{c} 2w_{ij}[J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}) - t^*_{ij}] \frac{\partial}{\partial \mathbf{B}} J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B}), \tag{23}$$

and η_1 and η_2 are the step lengths for the gradient ascent (descent) procedures. The closed form expression for $(\partial/\partial\mathbf{B})J_{KL}(\mathcal{N}_i,\mathcal{N}_j;\mathbf{B})$ is [34]

$$[\mathbf{S}_i\mathbf{B} - \boldsymbol{\Sigma}_i\mathbf{B}(\mathbf{B}^\top\boldsymbol{\Sigma}_i\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{S}_i\mathbf{B}](\mathbf{B}^\top\boldsymbol{\Sigma}_i\mathbf{B})^{-1}$$
$$+ [\mathbf{S}_j\mathbf{B} - \boldsymbol{\Sigma}_j\mathbf{B}(\mathbf{B}^\top\boldsymbol{\Sigma}_j\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{S}_j\mathbf{B}](\mathbf{B}^\top\boldsymbol{\Sigma}_j\mathbf{B})^{-1}, \tag{24}$$

where $\mathbf{S}_i = \boldsymbol{\Sigma}_j + \mathbf{U}_{ij}$ from (9), and $\mathbf{S}_j = \boldsymbol{\Sigma}_i + \mathbf{U}_{ij}$.

The step length parameters, η_1 and η_2, are initially small (0.01 in all our experiments) and they are decreased by a factor of 50% if the objectives $\mathcal{E}_{PDAWS}(\mathbf{B})$ and $\mathcal{L}_{PDALP}(\mathbf{B})$ decrease (instead of increase) or increase (instead of decrease). Other strategies such as explicit line search are possible but this simple method has provided good results in all our experiments. Since the gradient ascent (or descent) method can be trapped into local minima, the algorithm is restarted with multiple initializations (10 times in all our experiments) and the solution with the lowest error on the training (or validation) set is selected as the final solution. Alternatively, $\mathbf{B}^*$ can be selected using cross-validation.

The optimal solution $\mathbf{B}^*$ obtained from (13) and (14) is a local optimum for $\mathcal{E}_{PDAWS}(\mathbf{B})$ and $\mathcal{L}_{PDALP}(\mathbf{B})$, respectively. Note that $\mathbf{B}^*$ is declared an optimal solution when the gradient is zero and there is no change in the objective function. From (3.1) and (3.3), $\mathbf{B}^*$ is *Pareto Optimal* and by definition, it is one of the solutions in the Pareto front.

Discussion: The WS method is in the same spirit of MODA, albeit it assigns the objective function for each pair of classes $J_{KL}(\mathcal{N}_i,\mathcal{N}_j)$ a weight w_{ij}. If the weights are properly set such that overlapping classes have large weights, and well separated classes have a small weight, then the WS method will counteract the class masking effect. Note that the gradient for the WS method is also a weighted combination of the gradient from each objective function in (21). Hence, the total gradient will be dominated by gradient directions that separate between overlapping classes.

The L_p-Metric method, on the other hand, is different from the WS method and MODA/LDA as well due to the targets t_{ij} which act as constraints on the minimum divergence that each objective function must achieve. If t_{ij} is large enough $\forall i,j$ to ensure that all classes are well separated from each other, then by minimizing the difference $J_{KL}(\mathcal{N}_i,\mathcal{N}_j) - t_{ij}$, the L_p-Metric method will try to maximize the separation between classes such that it gets as close as possible to the desired t_{ij}. Here, the weights w_{ij} play a slightly different role for each objective function. If $J_{KL}(\mathcal{N}_i,\mathcal{N}_j)$ is close to t_{ij}, the corresponding w_{ij} should be small, while if $J_{KL}(\mathcal{N}_i,\mathcal{N}_j)$ is far from t_{ij}, then the corresponding w_{ij} should be large. In this way, the total gradient in (23) will be dominated by gradient directions for objective functions that are far away from the minimum desired KL divergence t_{ij}. Note that the WS method, similar to LDA/MODA, does not impose any constraints on the minimum divergence between classes.

[6] For the L_p-Metric method in Problem (5), p was set to 2 according to Corollary 3.3.

4.5. Initial bases $\mathbf{B}_0$

In order for the gradient ascent and descent procedures in (20) and (22) to find a good and stable local solution $\mathbf{B}^*$, a good initial bases $\mathbf{B}_0$ is needed to start the procedures. Such an initial bases can, for instance, keep all the discriminatory information in the data, and discard all dimensions with small or constant variance. This suggests that a good initial bases can be the bases obtained from principal component analysis (PCA), i.e., the columns of $\mathbf{B}_0$ are the d_0 largest eigenvectors of the total scatter matrix $\mathbf{S}_t$, since the null space of $\mathbf{S}_t$ contains no useful discriminatory information in the data [14,35]. Note that for zero mean data, the maximum rank for $\mathbf{S}_t$ is $\min(d-1, n-1)$, and hence this allows $\mathbf{B}$ to be of dimension $d \times d_0$, where $1 \leq d_0 \leq \min(d-1, n-1)$.

Another possible option is to use the first few eigenvectors of the between-class scatter matrix $\mathbf{S}_b$. This option, however, has two limitations: (i) it will bias PARDA towards the LDA solution which already suffers from the class masking problem, and (ii) the maximum rank of $\mathbf{S}_b$ is $\min(c-1, d-1)$, and hence the dimensionality d_0 of the embedding space will be limited to the rank of $\mathbf{S}_b$. For this reason, we opted for the first option in all our experiments.

4.6. The target vector $\mathbf{t}^*$

Before discussing the selection of the weights w_{ij}, it is important to discuss first how to set the targets t_{ij} for the L_p-Metric method in (14). Selecting the weights w_{ij} for the WS and for L_p-Metric methods will rely on the targets t_{ij}. The rational for selecting the target values as discussed below rests on our understanding for the class masking problem (Section 4.1), the information loss due to the linear dimensionality reduction (Section 4.3), the ideal (or utopia) point in the L_p-Metric method (Section 3.2), and the optimal separation between Gaussian densities for learning mixtures of Gaussians [36].

Recall that an ideal solution would be a low dimensional subspace in which all class means are maximally separated, while the class spread is minimized. Since the targets t_{ij} in (14) are constraints on the desired minimum interclass divergences, it is possible to encourage such an ideal solution by setting all t_{ij}'s to be equal to one large value t^*. A large enough value for t^* will encourage solutions in which the class means are far from each other as much as possible (while minimizing the overlap between classes due to class scatter), and setting $t_{ij} = t^*, \forall i,j$, will encourage solutions in which all class means are equally distant from each other, while class overlap is minimized. However, how large should be t^*? For the purpose of the following discussion only, we will be dealing with the transformed Gaussians $\mathcal{N}_j(\mathbf{B}_0^\top \boldsymbol{\mu}_j,\ \mathbf{B}_0^\top \boldsymbol{\Sigma}_j \mathbf{B}_0),\ 1 \leq j \leq c$, in $\mathbb{R}^{d_0}$ obtained from $\mathbf{B}_0$. To reduce notations' cumbersomeness, $\mathbf{B}_0$ will be omitted from the notation.

An early result on learning mixtures of Gaussians (MOG) [36] states that two Gaussian densities, $\mathcal{N}_1$ and $\mathcal{N}_2$, are said to be τ–separated in $\mathbb{R}^{d_0}$ if

$$\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 > \tau \sqrt{d_0 \max(\lambda_{\max}(\boldsymbol{\Sigma}_1), \lambda_{\max}(\boldsymbol{\Sigma}_2))}, \tag{25}$$

where $\lambda_{\max}(\boldsymbol{\Sigma})$ is the largest eigenvalue of matrix $\boldsymbol{\Sigma}$. Further, a MOG is τ-separated if its Gaussian components are pairwise τ-separated. In high dimensions, a 2-separated MOG is almost a completely separated set of Gaussians [36]. Setting $\tau = 2$ and squaring the left and right terms of (25) we get

$$\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2^2 > 4 d_0 \lambda^* \quad \text{where } \lambda^* = \max_{1 \leq j \leq c} \lambda_{\max}(\boldsymbol{\Sigma}_j),$$

which is the minimum squared distance between any two means $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ to ensure separation. Hence, a reasonable value for t^* would be

$$t^* = 4 d_0 \lambda^*. \tag{26}$$

Note that τ can be slightly bigger for very high dimensional data, and in practice it can be selected using cross validation. In all our experiments however, we used $\tau = 2$. Recall that t^* is defined in the lower d_0 dimensional subspace defined by the initial bases $\mathbf{B}_0$, and hence t^* is in fact a function of $\mathbf{B}_0$. Based on the value of t^*, it is worth noting the following. First, t^* could have been computed in the original input space $\mathbb{R}^d$, however due to the information loss incurred from $\mathbf{B}_0$ (Section 4.3), such a value will be huge with respect to all the pairwise divergences $J_{KL}(\mathcal{N}_i, \mathcal{N}_j; \mathbf{B})$, and will mask all the differences among them. This is unlike t^* in (26) which is a function of $\mathbf{B}_0$, and hence it is suitably large enough for all the divergences $J_{KL}(\mathcal{N}_i, \mathcal{N}_j; \mathbf{B})$, and scales reasonably with them. Second, it is not expected that any objective function $J_{KL}(\mathcal{N}_i, \mathcal{N}_j; \mathbf{B}),\ \forall i,j$, will ever attain t^*, rather it only encourages the optimization procedure to search for solutions where class means are maximally separated, while the overlap between classes due to class scatter is minimized. Last, this unachievable value for t^* coincides with the role of the ideal point in the L_p-Metric method which happens to be meaningful and useful for DA.

4.7. The weights w_{ij}

The weights w_{ij} play a crucial role for the WS and the L_p-Metric methods since they drive the optimization procedure to concentrate on more important objective functions (with large weights) in favour of other less important ones (with small weights). Recall that the gradients in (21) and (23) involve all w_{ij}'s.

To encourage the optimization procedure to achieve the ideal DA setting, we will rely on the target t^* and the initial bases $\mathbf{B}_0$ in order to set all the weights w_{ij}. If $J_{KL}(\mathcal{N}_i, \mathcal{N}_j; \mathbf{B}_0)$ is close to t^*, it is expected that classes $\mathcal{N}_i$ and $\mathcal{N}_j$ are well separated. Hence, the optimization procedure should consider minimal effort to further maximize $J_{KL}(\mathcal{N}_i, \mathcal{N}_j; \mathbf{B}_0)$. In the contrary, if $J_{KL}(\mathcal{N}_i, \mathcal{N}_j; \mathbf{B}_0) \ll t^*$, it is expected that $\mathcal{N}_i$ and $\mathcal{N}_j$ are close to each other. In this case, the optimization procedure should put more effort to increase the separation between these two classes, while at the same time, prevent all other classes from overlapping over each other. To achieve such a mechanism, we set the weights w_{ij} as follows:

$$w_{ij} = \frac{\delta_{ij}}{\sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \delta_{ij}} \quad \text{and} \quad \delta_{ij} = \frac{t^*}{J_{KL}(\mathcal{N}_i, \mathcal{N}_j; \mathbf{B}_0)}. \tag{27}$$

Under this weighting scheme, the smaller the separation (or divergence) between $\mathcal{N}_i$ and $\mathcal{N}_j$, the larger the δ_{ij} will be, and hence the larger is the weight w_{ij}, i.e., a more important objective function since it is far away from t^*. In contrast, the larger the divergence between $\mathcal{N}_i$ and $\mathcal{N}_j$, the smaller the δ_{ij} will be, and hence the smaller is the weight w_{ij}, i.e., less important objective function since it is closer to t^*. Note that this weighting scheme does not differentiate between the WS and the L_p-Metric methods, and hence it is used for both of them. To see how this weighting mechanism works against the class masking effect, consider the gradient for the WS and L_p-Metric methods in (23) and (21), respectively. The right-hand side of (23), for instance, is a weighted convex combination of all the gradients from each objective function $J_{KL}(\mathcal{N}_i, \mathcal{N}_j; \mathbf{B}),\ \forall i,j$. If the objective function is close to t^*, the corresponding weight w_{ij} will be small, and hence the contribution of this gradient to the total gradient will be minimal. Vice versa, when the objective function is far away from the minimum desired separation t^*, the corresponding weight w_{ij} will be large, and hence this gradient direction will have a major contribution to the total gradient. That is, the total gradient is dominated by gradient directions for those objective functions that are far away from t^* since they contribute with large weights. Note that a similar interpretation follows for the gradient of the WS method. This is, however, the opposite of LDA/MODA in which the total gradient is dominated by the largest symmetric KL divergence (i.e. the most distant pair of classes).

The weights w_{ij} assigned to the multiobjective function (WS or L_P-Metric) at the beginning of the optimization procedure are fixed and do not change during the iterative procedure. The reason for that has to do with convergence.[7] If the weights are changed in each iteration, each set of weights will define a new optimization problem in each gradient ascent (descent) step. Consequently, the sequence of gradient vectors will point to different directions in the objectives space and will not converge to a solution. However, having adaptive weights at each gradient step is indeed possible assuming that the weights updating rule will result in consistent gradient directions towards the optimal solution.

4.8. Computational complexity

Maximizing $\mathcal{E}_{PDAWS}$ in (13) and minimizing $\mathcal{L}_{PDALP}$ in (14) rely on the iterative gradient ascent and descent procedures in (20) and (22), respectively. A sensible quantity for PARDA's time complexity is the complexity of each iteration in these procedures. Each iteration is dominated by two main operations: (i) Eq. (10) which evaluates the KL divergence between two Gaussian densities, and (ii) Eq. (24) which evaluates the expression for the gradient.

The time complexity for evaluating the KL divergence in (10) is $O(\epsilon d_0^3+\epsilon_1 d_0^2 d)$ in the worst case. The cubic term is due to the matrix inversions and it occurs in the d_0-dimensional subspace (where $d_0 \ll d$). Similarly, the time complexity for the gradient expression in (24) is $O(\epsilon d_0^3+\epsilon_2 d_0^2 d)$ in the worst case. The cubic term is also due to the matrix inversion in the d_0-dimensional subspace. The term $d_0^2 d$ is due to matrix vector multiplications. These complexities, however, are for one pair of classes, and for a c-class problem, the complexity is $O\Big((c(c-1)/2)(\epsilon d_0^3+\epsilon_1 d_0^2 d)\Big)$ for Eq. (10), and $O\Big((c(c-1)/2)(\epsilon d_0^3+\epsilon_2 d_0^2 d)\Big)$ for Eq. (24), in the worst case, where $1<\epsilon,\epsilon_1,\epsilon_2 \in \mathbb{N}_+$. Note that these complexities are independent of the number of samples n, linear in the number of input features d, and quadratic in the number of classes c. Note also that, at each iteration, the gradient expression is evaluated once, and hence, searching for the optimal step size η is dominated by evaluating the KL divergence in the low dimensional subspace.

4.9. Relation to other approaches

We consider here the differences between the proposed PARDA models and some other approaches for the class masking problem. Unlike aPAC [11] and similar methods that focus on better estimates for the between-class scatter matrix $\mathbf{S}_b$, Tang et al. [31] propose a better estimate for the within-class scatter matrix $\mathbf{S}_w$ to be used within the aPAC algorithm. In their approach, $\mathbf{S}_w$ is a weighted average of classes' covariance matrices: $\mathbf{S}_w=\sum_{j=1}^{c} p_j r_j \boldsymbol{\Sigma}_j$, where p_j is the *a priori* probability for class L_j, and r_j is a measure of separability between class L_j and all other classes in the original input space. If L_j is already well separated from all other classes, then r_j will be small. This method, however, relies on separability information from the original input space which is avoided in PARDA as discussed in Section 4.6. Further, similar to aPAC, these methods do not provide an explicit mechanism that encourages a minimum separation between the classes in the lower dimensional subspace.

Bian and Tao [32] propose a more direct approach for the class masking problem, namely mini–max distance analysis (MMDA) for dimensionality reduction. MMDA searches for a bases matrix $\mathbf{B}$ that maximizes the pairwise squared Euclidean distance between the class means in the homoscedastic Gaussian setting. For the heteroscedastic Gaussian setting, they extend their model using the kernel trick. MMDA's original objective function is non-smooth, and hence the authors lean to a relaxed objective function as an approximation to the original one, and then solve it by sequences of semi-definite programs (SDPs). Similar to PARDA, their approach encourages a minimum separation (in Euclidean distance sense) between classes in the lower dimensional space. PARDA, on the other hand, directly considers a heteroscedastic Gaussian setting and maximizes the separation between classes by means of a discrimination measure, the symmetric KL divergence, which takes the means and covariance matrices into consideration. Note that the approximated problem heavily relies on the initial solution since the final solution will be around this initial point. Also, it is not clear how the sequence of relaxed formulations that MMDA is solving affects the final discriminating subspace. In addition, the sequence of SDPs is solved via an interior point method which, despite all recent advances, is still not computationally attractive.[8]

Finally, the weights proposed in the aPAC algorithm for $\mathbf{S}_b$, and the weights r_j proposed in [31], can be investigated as alternatives for, or complimentary to the weights w_{ij} based on the τ-separated MOG [36] proposed here. Other direct approaches based on cost sensitive classification can also be used in PARDA [38].

5. Extension to multimodal classes

In this section we extend PARDA to the realistic setting in which the class distributions are non-Gaussian and multimodal. As it will be shown, the model will be able to handle this setting in a smooth manner with almost no changes to any of the details presented earlier.

To handle multimodal class distributions, it is reasonable to approximate each class distribution as a mixture of Gaussians (MOG) [23,15,13]. However, there are two main usual concerns associated with such an approximation: (i) which clustering algorithm to use, and (ii) the number of mixture components for each class. A simple, yet an efficient solution is to use the k-Means clustering algorithm (with multiple initializations) to cluster each class into h subsclasses. Zhu and Martinez [13] studied two criteria for optimal subclass division, and proposed the nearest neighbour (NN) clustering algorithm to divide a class into h subclasses. Alternatively, De La Torre and Kanade [15] used multiway normalized cuts [39] for the same purpose. For all these approaches, the number of subclasses h can be empirically chosen to minimize the training or validation error, or chosen using cross validation. Once each class L_j is divided into h_j subclasses, each subclass is modelled as Gaussian distribution $\mathcal{N}_{jk}$, where $1 \le j \le c$, and $1 \le k \le h_j$. This turns the optimization problem for PDAWS to be

$$\mathbf{B}^*_{PDAWS} = \underset{\mathbf{B}\in\mathcal{R}}{\operatorname{argmax}}\ \mathcal{E}_{PDAWS}(\mathbf{B}),$$
$$\text{s.t.} \sum_{i,j,k_1,k_2} w_{ik_1jk_2}=1, \quad w_{ik_1jk_2}>0 \quad \text{where} \tag{28}$$

$$\mathcal{E}_{PDAWS}(\mathbf{B}) = \sum_{i=1}^{c-1}\sum_{j=i+1}^{c}\left(\sum_{k_1}^{h_i}\sum_{k_2}^{h_j} w_{ik_1jk_2} J_{KL}(\mathcal{N}_{ik_1},\mathcal{N}_{jk_2};\mathbf{B})\right). \tag{29}$$

For PDALP the optimization problem becomes

$$\mathbf{B}^*_{PDALP} = \underset{\mathbf{B}\in\mathcal{R}}{\operatorname{argmin}}\ \mathcal{L}_{PDALP}(\mathbf{B}),$$
$$s.t. \sum_{i,j,k_1,k_2} w_{ik_1jk_2}=1, \quad w_{ik_1jk_2}>0 \quad \text{where} \tag{30}$$

$$\mathcal{L}_{PDALP}(\mathbf{B}) = \sum_{i=1}^{c-1}\sum_{j=i+1}^{c}\left(\sum_{k_1}^{h_i}\sum_{k_2}^{h_j} w_{ik_1jk_2}[J_{KL}(\mathcal{N}_{ik_1},\mathcal{N}_{jk_2};\mathbf{B})-t^*]^2\right). \tag{31}$$

To understand how the formulation in (29) and (31) handle the multimodal setting, it is important to note that index j is always

[7] Personal communication with M. Ehrgott [22].

[8] Currently, there is a surge in solving SDPs by means of Frank–Wolfe type algorithms, a.k.a first order methods, or conditional gradient methods. See for instance [37].

greater than index i, and hence there is no weight $w_{ik_1jk_2}$ in which $i=j$. That is, the set of weights exists only between subclasses from different classes, and zero otherwise. Note also that the gradients for $\mathcal{E}_{PDAWS}$ and $\mathcal{L}_{PDALP}$ in (21) and (23) will include the weights $w_{ik_1jk_2}$. Given how the weights are set based on the target t^*, PARDA will be encouraged to search for subspaces in which subclasses from different classes are separated from each other. Note that the weights between classes are now replaced by the weights between subclasses from different classes.

6. Empirical analysis

In this section we carry out different types of experiments using PARDA to compare its performance to that of well known and modern DA algorithms. Two types of data sets were used in all our experiments: synthetic and real. The synthetic data set, referred here as `Gaussians`, has 5 classes, each has a 20 dimensional Gaussian distribution with a full covariance matrix. The data set has 1000 samples for training (200 samples/class), and 1000 samples for test (200 samples/class). The data set was generated as follows [15]. Each sample $\mathbf{x}_i$ from class L_j is generated as $\mathbf{x}_i = \mathbf{T}_j\mathbf{b}+\boldsymbol{\mu}_j+\mathbf{n}$, where $\mathbf{x}_i \in \mathbb{R}^{20}$, $\mathbf{T}_j \in \mathbb{R}^{20\times 7}$, $\mathbf{b} \sim \mathcal{N}_7(0,\mathbf{I})$, $\mathbf{n} \sim \mathcal{N}_{20}(0,2\mathbf{I})$ for training samples, and $\mathbf{n} \sim \mathcal{N}_{20}(0,2.3\mathbf{I})$ for test samples. The bases $\mathbf{T}_j$ are random matrices where each element is generated from $\mathcal{N}(0,5)$. Further, the bases $\mathbf{T}_j$ are enforced to be orthogonal to each other; that is $\mathrm{tr}(\mathbf{T}_i\mathbf{T}_j^\top)=0, \forall i\neq j$. This can be achieved using the following Gram–Schmidt approach: $\mathbf{T}_j = \mathbf{T}_j - \sum_{k=1}^{j-1}\mathrm{tr}\{(\mathbf{T}_k\mathbf{T}_k^\top)^{-1}\mathbf{T}_j^\top\mathbf{T}_k\}\mathbf{T}_k$, for $j=2,\ldots,5$. The means of each class were as follows: $\boldsymbol{\mu}_1 = 4\mathbf{1}_{20}$, $\boldsymbol{\mu}_2 = \mathbf{0}_{20}$, $\boldsymbol{\mu}_3 = -4[\mathbf{0}_{10},\mathbf{1}_{10}]^\top$, $\boldsymbol{\mu}_4 = 4[\mathbf{1}_{10},\mathbf{0}_{10}]^\top$, and $\boldsymbol{\mu}_5 = 4[\mathbf{1}_5,\mathbf{0}_5,\mathbf{1}_5,\mathbf{0}_5]^\top$.

The real data sets, shown in Table 1, were selected from various domain knowledge to verify whether PARDA can generalize on a large variety of problems. All data sets were selected such that they have more than two classes: (i) Nine data sets from the UCI machine learning repository [40]; `newthyroid` (for illustration purposes only), `isolet`, `optdigits`, `pageblocks`, `pendigits`, `satimages`, `segment`, `shuttle`, and `vowel`, (ii) The Ohio sitting posture (`osp`) data set [41], (iii) The `usps` [42] and `mnist` [43] data sets, and (iv) Two faces data sets: `cmupie` [44] & `yaleb` [45].

For comparisons with PARDA, the following algorithms were selected from the literature: Principal Component Analysis (PCA), White+LDA [46] which is known to work well in practice for face recognition problems, Approximate Pairwise Accuracy Criterion LDA (aPAC)[9] [11], Subclass Discriminant Analysis (SDA)[10] [13], Relevant Component Analysis (RCA) [47], and MODA [15]. Note that RCA was proposed as an algorithm for learning a Mahalanobis metric over data sets with side information (equivalence constraints in this case). When provided with a fully labeled data set, and noticing its algorithmic implementation, it is equivalent to performing PCA followed by LDA.

The performance measure in our experiments is the classification error of a quadratic discriminant analysis (QDA) classifier in the low dimensional subspace. That is, a sample $\mathbf{x}$ in the input space is projected into the low dimensional subspace using the low rank matrix $\mathbf{B}$ obtained from the Pareto model or from a competitive algorithm, $\mathbf{B}^\top\mathbf{x}=\mathbf{y}$, and then $\mathbf{y}$ is assigned to the class that yields the smallest discriminant score: $(\mathbf{y}-\widehat{\boldsymbol{\mu}}_j)^\top\widehat{\boldsymbol{\Sigma}}_j(\mathbf{y}-\widehat{\boldsymbol{\mu}}_j)+\log|\widehat{\boldsymbol{\Sigma}}_j|-2\log\widehat{\pi}_j,\ \forall j$, where $\widehat{\boldsymbol{\mu}}_j$ and $\widehat{\boldsymbol{\Sigma}}_j$ are, respectively, the low dimensional estimates for the mean vector and covariance matrix for class L_j, and $\widehat{\pi}_j$ is the empirical class prior probability.

[9] From the PRTools found at: `prtools.org`.
[10] From the authors' website.

Table 1
Data sets used in our experiments with their size (n), number of features (d), and number of classes (c).

Name	n	d	c
`uci newthyroid`	215	5	3
`uci isolet`	7797	617	26
`uci pageblocks`	5473	10	5
`uci satimages`	6435	36	6
`uci segment`	2310	18	7
`uci shuttle`	58,000	9	7
`uci vowel`	990	11	11
`digits usps`	9298	16 × 16	10
`faces cmupie`	11,554	32 × 32	68
`faces yaleb`	2414	32 × 32	38
`OSP`	2500	1080	10
`digits mnist`	60–10 K	24 × 24	10

6.1. Experimental results

In our first experiment, we consider the projection quality obtained from PARDA and how it compares to the projections obtained from other algorithms. Figs. 3 and 4 show the 2D projections for `Gaussians` and `newthyroid`, respectively. Note that for LDA, the maximum d_0 is $c-1$, which for the two data sets are $d_0=4$ and $d_0=2$, respectively. In each figure, the projections are obtained using PCA, WLDA, aPAC, RCA, MODA, PDAWS, and PDALP.

For the `Gaussians` case, Fig. 3, all algorithms give very similar projections. Note that the class compactness, class overlap, relative location of classes to each other, and the classification error are similar in all cases, except for the error resulting from the PCA projection. This is understandable since PCA projects the data on the directions with maximum variance irrespective of the class labels. For `newthyroid`, Fig. 4, one can notice the difference between PDALP on one hand, and all other methods on the other hand. PDALP yields projections with less overlap than all other methods and resulted in the lowest error rate. This is due to the mechanism of the target values t_{ij} and the weights w_{ij}. To see this, note that PDAWS (which does have similar weights but without target values) resulted in higher error rate than PDALP. MODA, which does not have weights or target values, resulted in higher overlap between classes and higher error rate. RCA and aPAC yielded very similar projections and resulted in the same error rate. Although both algorithms performed better than PDAWS in this case, Tables 2–4, will show various cases where PDAWS outperforms these two algorithms.

In our second experiment, we consider the behaviour for the training error and test error when increasing the number of bases d_0 for the low rank projection matrix $\mathbf{B}$ (i.e. adding more features), especially when $d_0 > c-1$. Fig. 5a and b shows the average training and test errors (with standard deviations) versus the number of bases d_0 for `Gaussians` and `optidigits`, respectively. The average errors were measured as follows. First, we generated 10 different instances from the `Gaussians` data set described earlier. The final training and test errors were the average error over the 10 training sets and 10 test set errors, respectively. Second, for `optidigits`, the average training and test errors were obtained using 10 folds double cross validation.

For the `Gaussians` case, the training error decreases as more dimensions (or features) are added to the data representation. The test error, on the other hand, starts high when $d_0 < c-1$, minimum at $d_0 = c-1$, and then increases as more dimensions (features) are added. These profiles, especially for the average test error, are the typical profiles for error vs. model's complexity of a learning algorithm when the data set is fixed [48, p. 194]. Note that the

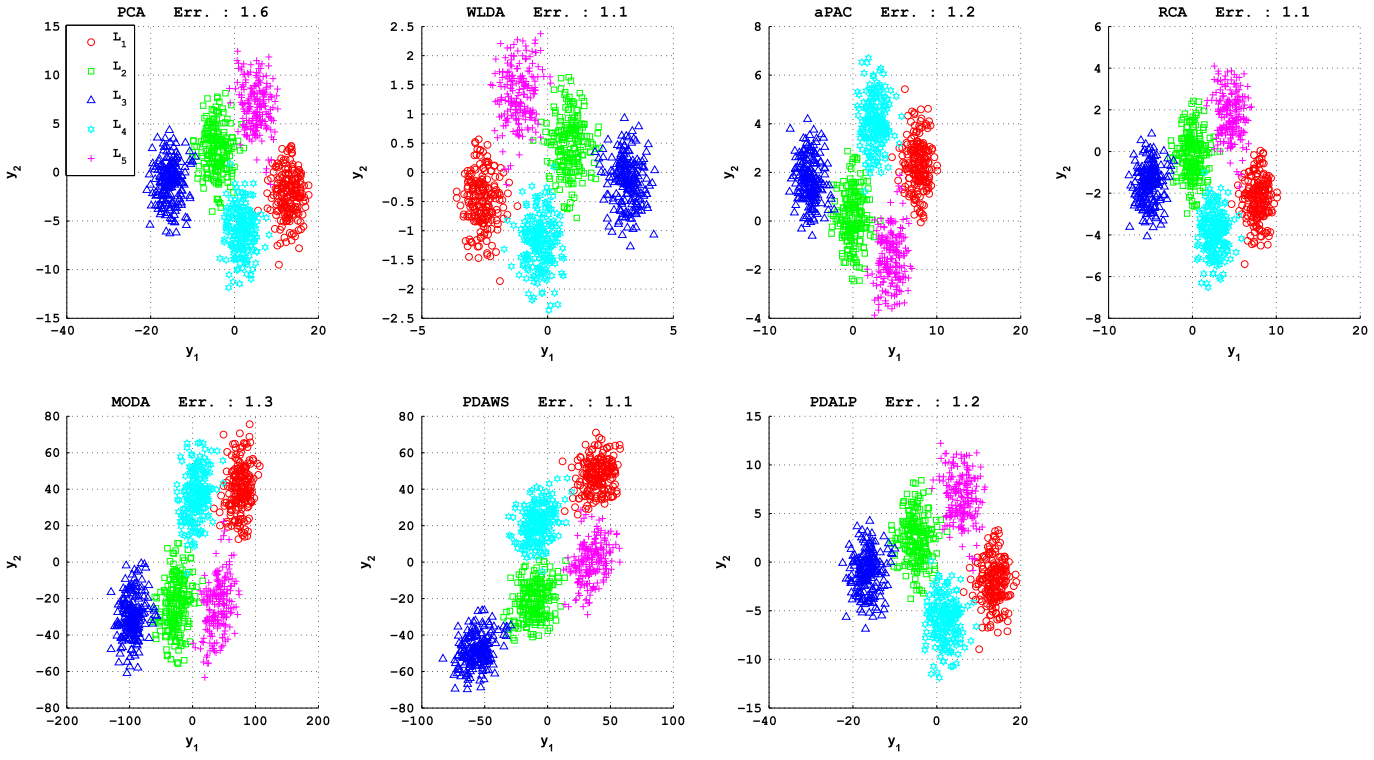

Fig. 3. 2D projections for `Gaussians` with training error using PCA, WLDA, aPAC, RCA, MODA, PDAWS, and PDALP.

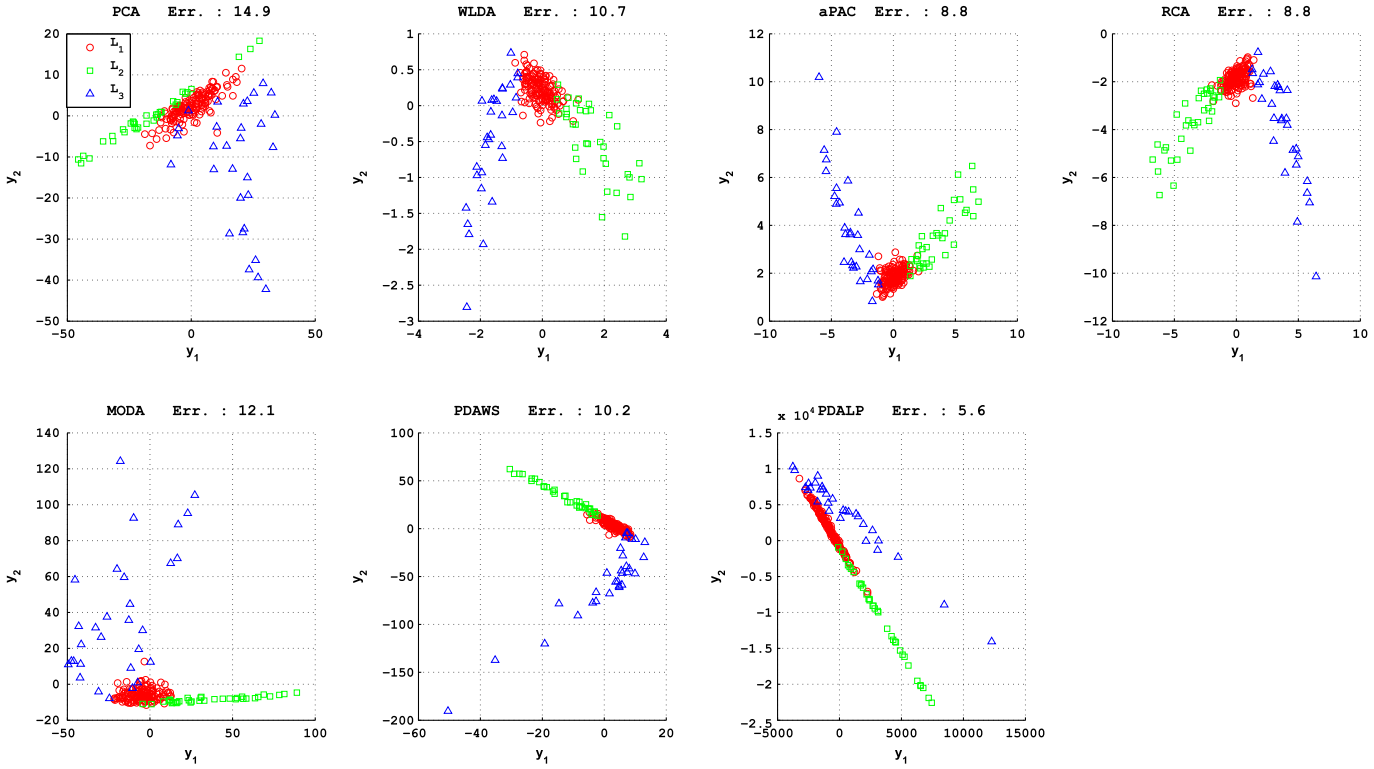

Fig. 4. 2D projections for `newthyroid` with training error using PCA, WLDA, aPAC, RCA, MODA, PDAWS, and PDALP.

model's complexity here refers to the complexity of the representation (or features), which is represented here by adding more bases (or columns) to matrix **B**. Hence, PARDA behaves normally in that regard.

In our last experiments, the PARDA models are compared with the six algorithms (mentioned earlier) on the 13 real data sets in Table 1. Four different PARDA based algorithms are included in the comparison: (i) The L_p-Metric method (PDALP), (ii) The WS method (PDAWS), (iii) The L_p-Metric method with each class modelled as a MOG with two components (MOGLP), and (iv) The WS method with each class modelled as a MOG with two components (MOGWS). For PARDA models, the number of dimensions d_0 can be selected to

Table 2
Average test error rates (with standard deviation) for all algorithms on the nine UCI data sets and the usps data set. Default $d_0 = \min(c-1, d-1)$. Numbers in square brackets indicate d_0's value when it is different from the default value.

Data set	d	QDA	d_0	PCA	WLDA	aPAC	RCA	SDA	MODA	PDALP	PDAWS	MOGLP	MOGWS
pendigits	16	**1.8 (0.5)**	9	5.2 (0.7)	6.9 (1.4)	7.0 (1.5)	7.0 (1.5)	6.9 (1.5)	5.1 (0.7)	3.4 (0.4)	4.2 (0.5)	4.3 (0.6)	4.2 (0.5)
satimages	36	15.8 (3.8)	5	17.5 (5.2)	16.9 (4.8)	17.4 (4.9)	17.4 (4.8)	18.2 (5.4)	16.9 (4.9)	**14.0 (4.5) [4]**	**14.8 (3.8)**	**14.1 (4.0)**	**14.8 (4.3)**
segment	18	11.1 (2.3)	6	22.2 (1.7)	7.6 (1.8)	7.6 (1.8)	8.1 (1.8)	**7.2 (1.9)**	10.2 (2.3)	**7.1 (2.3)**	**7.5 (1.2)**	**6.9 (1.7)**	**7.5 (2.4) [10]**
shuttle	9	5.4 (0.8)	6	4.3 (0.3)	19.2 (3.6)	6.2 (0.4)	6.2 (0.4)	4.3 (0.2)	4.0 (0.3)	**3.4 (0.6) [5]**	**3.6 (0.3)**	**3.6 (0.3)**	**2.6 (0.3)**
vowel	11	36.4 (7.7)	10	35.8 (8.4)	42.1 (8.7)	42.2 (8.8)	42.2 (8.8)	40.2 (8.6)	38.5 (8.1)	**34.9 (9.1)**	**33.1 (8.4) [5]**	**33.7 (9.1)**	**35.8 (8.8) [9]**
isolet	617	14.3 (2.4)	25	9.2 (2.8)	**6.1 (1.7)**	**6.2 (1.8)**	**6.2 (1.8)**	**6.3 (1.8)**	13.8 (3.5)	**5.9 (2.1)**	8.6 (2.1)	**6.5 (1.7)**	6.9 (1.7)
optdigits	64	**3.8 (0.7)**	9	5.0 (1.1)	**3.9 (1.2)**	**3.9 (1.2)**	**3.9 (1.1)**	**3.9 (1.1)** **1.9 (0.6) [16]**	5.5 (1.4) **2.2 (0.8) [20]**	**3.5 (0.8)** **1.8 (0.6) [19]**	4.7 (1.2) **2.6 (0.7) [20]**	**3.7 (1.0)**	5.2 (1.2)
pageblocks	10	**6.8 (3.5)**	4	67.9 (12.9)	25.4 (12.9)	43.5 (15.2)	43.9 (15.2)	54.0 (19.0)	55.8 (14.2)	8.92 (3.4) [3] **6.3 (10.6) [7]**	7.8 (3.2) [3]	8.4 (3.4)	9.9 (5.6)
usps	256	7.4 (4.0)	9	11.3 (1.3)	**6.91 (1.4)**	7.2 (1.5)	7.2 (1.5)	7.1 (1.5) **3.2 (0.9) [19]**	19.4 (1.2) **3.6 (0.8) [20]**	**5.4 (1.3)** **3.8 (1.0) [17]**	7.3 (1.1) **4.2 (1.3) [19]**	**5.5 (1.0)**	8.9 (1.2)

Table 3
Average test error rates (with standard deviation) for all algorithms on cmupie, yaleb, and osp data sets. Default $d_0 = \min(c-1, d-1)$. Numbers in square brackets indicate d_0's value when it is different from the default value.

Data set	d	QDA	d_0	PCA	WLDA	aPAC	RCA	SDA	MODA	PDALP	PDAWS	MOGLP	MOGWS
cmupie	1024	9.1 (17.7)	67	**7.7 (17.1)**	13.8 (19.9)	15.7 (20.3)	15.7 (20.3)	17.1 (20.6)	10.4 (18.5)	**7.6 (17.1)**	**7.5 (16.9)**	**7.8 (17.0)**	**7.5 (16.9)**
yaleb	1024	54.5 (14.3)	37	**6.6 (10.1)**	13.1 (16.1)	14.2 (16.3)	14.2 (16.3)	25.5 (19.5)	20.1 (15.8)	**5.8 (9.5)**	**5.6 (8.8)**	**6.9 (11.1)**	**6.5 (9.6)**
osp	1080	57.5 (5.5)	9	**31.1 (8.6)**	45.9 (8.5)	46.7 (8.5)	46.7 (8.5)	45.8 (7.9)	39.2 (6.3) **30.7 (9.0) [13]**	**31.0 (8.1)**	32.5 (9.1)	**29.6 (6.9)**	**29.7 (7.4)**

Table 4
Test error rates for all algorithms on the mnist data set. Default $d_0 = \min(c-1, d-1)$. Numbers in square brackets indicate d_0's value when it is different from the default value.

Data set	d	QDA	d_0	PCA	WLDA	aPAC	RCA	SDA	MODA	PDALP	PDAWS	MOGLP	MOGWS
mnist	576	12.6	9	11.4	12.1	12.7	12.6	12.6	24.1	**8.9** **3.9 [31]**	**9.1** **4.1 [35]**	**4.8 [19]**	**4.6 [19]**

minimize the training error or the validation error, or by means of cross validation for the number of samples is small. For MOGLP and MOGWS, note that the number of classes is doubled and the discriminant subspace can have a dimensionality up to $2c-1$, assuming $d > 2c$. Although the number of Gaussians per class is fixed for all data sets, a better approach would be to optimize the number of components per data set. Nevertheless, we obtained satisfactory results using this setting.

For all data sets, the performance measure is the test error of a QDA classifier. Column three in Tables 2 and 3 shows the average test error rates (with standard deviation) for the QDA classifier (using 10 folds cross validation) on the data sets before applying any dimensionality reduction. For convenience, column two in Tables 2–4 shows the number of feature d for each data set, while column four shows the value of $d_0 = \min(d-1, c-1)$.

Except mnist, columns 5–13 in Tables 2 and 3 show the average test error rates (with standard deviation) for the QDA classifier (using 10 folds cross validation) after applying the different dimensionality reduction algorithms. For the mnist data set, the training set (60 K samples) was split into two smaller sets: the first 40 K samples as a new training set, and the last 20 K samples as a validation set. All algorithms were trained on the new training set and used the validation set for parameter optimization. The results reported in Table 4 are the error rates for the QDA classifier on the mnist test set (10 K samples). Note that no preprocessing or feature extraction were applied to any of the images data sets (mnist, usps, cmupie, yaleb) – just raw vectorial data.

Table 2 shows the results for the UCI and USPS data sets. It can be seen that PARDA is consistently better or as good as other dimensionality reduction methods. For satimages, shuttle, vowel, and pageblocks, PARDA achieves the lowest error rates at fewer number of discriminant dimensions (shown in square brackets) than is required by other algorithms.

Table 3 shows the results for cmupie, yaleb, and osp data sets. In these cases PARDA outperforms WLDA, aPAC, RCA, SDA, and MODA. However, it can be noticed that PCA (which completely ignores the class labels) outperforms these algorithms as well and yields very competitive results to PARDA. This behaviour is not surprising and it has been shown in [49] that, if n_j, the number of samples per class, is much smaller than d, or if the training data is not uniformly sampled from its underlying manifold, PCA can outperform discriminant analysis methods. While these factors have hampered other DA methods, PARDA seemed to be robust against these factors and yielded competitive results. Note that MODA outperformed WLDA, aPAC, RCA, and SDA on these data sets but not as good as PCA for the same reasons. If taking the differences between MODA and PARDA into consideration, this shows that iterative methods for DA seem to be more robust against these factors.

Table 4 shows the results for all algorithms on the mnist data set. At $d_0 = c-1$, PDALP and PDAWS are better than all other algorithms, while MOGLP and MOGWS achieve the lowest error rate due to the modelling of each class as MOG. As d_0 slightly increases for PDALP and PDAWS, both algorithms achieve the lowest error rates.

The results for QDA without applying dimensionality reduction are worth discussing. In two cases (pendigits, pageblocks), QDA without dimensionality reduction resulted in the lowest error rates,

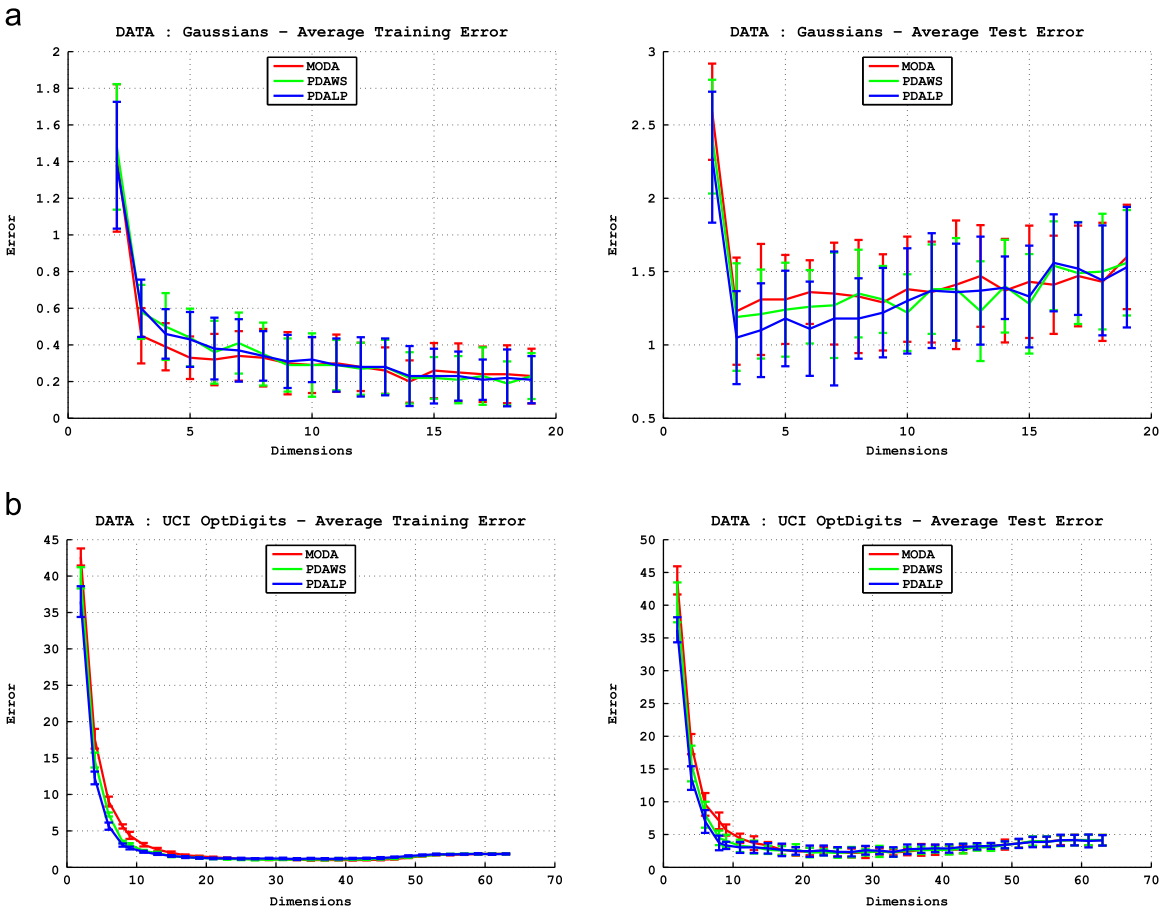

Fig. 5. Average training error (left) and average test error (right), with standard deviation, vs. d_0 using MODA, PDAWS, and PDALP, on (a) `Gaussians`, and (b) `optdigits`.

and only PARDA yielded the closest performance. If this is the case in a practical setting, then dimensionality reduction is not useful. However in all other cases, dimensionality reduction methods and PARDA, in particular, yielded subspaces with fewer dimensions and better discriminability between classes. This can be clearly seen in the cases of high dimensional data such as `isolet`, `optdigits`, `usps`, `cmupie`, `yaleb`, `osp` and `mnist`. Note that when the number of samples per class is much smaller than d (`yaleb`, `osp`), QDA had the worst performance. This demonstrates the crucial role of dimensionality reduction methods.

In general, the results above show that PARDA results in a promising performance with a sufficient room for improvements if one considers optimizing the dimensionality of the embedding subspace, and the number of mixture's components for each class. When comparing MODA to all PARDA based algorithms, one can notice how the convex weights try to counteract the class masking effect. Further, it can be noticed that PDALP and MOGLP yield better results than PDAWS and MOGWS. This is due to the target value t^* which plays a dual role: (i) it encourages subspaces with maximum separation between classes, and (ii) it acts as a constraint on the minimum divergence between classes.

7. Concluding remarks and outlook

We propose a new approach for supervised linear dimensionality reduction in the multiclass setting. In this approach, each class is modelled as a Gaussian distribution with a full covariance matrix, thereby casting the original problem into a multiclass heteroscedastic discriminant analysis (HDA) model. Unlike previous objective functions for HDA, our approach perceives the multiclass HDA as a set of functions, each representing the distance (in terms of symmetric KL divergence) between two different classes (or Gaussians). An ideal solution for such a set of functions is a low rank linear transformation matrix $\mathbf{B}^*$ that defines a low dimensional subspace in which all the pairwise distances between class means are maximized, and the overlap between classes due to class spread is minimized. Although such an ideal solution will not exist in practice, it raised the need for a mathematical model that can encourage such ideal solutions, and hopefully, attain an approximation for it.

The mathematical model turns to be the machinery of multi-objective optimization for which its optimal solution is known to exist. The Pareto optimal solution is suboptimal for each individual objective function, but is in maximal agreement between all the possibly conflicting objective functions. The proposed model, PARDA, concentrates on separating overlapping classes with an explicit mechanism to counteract the class masking effect. Experimental results on real data sets showed promising performance in favour of PARDA when compared with well known algorithms from the literature.

PARDA offers additional flexibility on two different aspects: parallel implementation for large scale settings, and using different measures of separation between classes. First, for problems with large number of classes, the iterative gradient ascent (or descent) procedure can be easily parallelized on today's multicore architectures, with minimal communications between the cores. To see this, note that in order to compute the gradient for one objective function, each core needs to have $\mathbf{B}^t$, $\boldsymbol{\mu}_i$, $\boldsymbol{\mu}_j$, and the regularized low rank estimates of $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_j$. Note also that computing $\mathcal{E}_{PDAWS}$ and $\mathcal{L}_{PDALP}$ can be parallelized in a similar fashion. Such a parallel implementation can offer a substantial speedup for the time required to learn the model.

Second, PARDA is not restricted to the symmetric KL divergence as a measure of separation between two classes. For instance, the Bhattacharyya distance and the Hellinger distance are well known symmetric divergence measures with closed form expressions for Gaussian densities. Using these divergence measures does not change any conceptual idea in the model, however, the expressions for the objective functions and the gradients will indeed change accordingly. Important questions in that regard are those related to the nature of the divergence measure, its capability as a measure of discrimination between classes, and its computational burden. A natural extension for this line of research is that of non-parametric DA within the PARDA model, and extending PARDA to operate in a kernel feature space.

Conflict of Interest

None declared.

Acknowledgements

This research was supported by the FQRNT–REPARTI Award for International Training (2009), FQRNT Postdoctoral Fellowship Award (2011–2013), and NSERC Discovery Grant Rgpin 36560–06.

Appendix A. Supplementary data

Supplementary data associated with this paper can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2014.11.008.

References

[1] R. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (1936) 179–188.
[2] K. Fukunaga (Ed.), Introduction to Statistical Pattern Recognition, Academic Press, Orlando, FL, 1972.
[3] C.R. Rao, Linear Statistical Inference and its Applications, John Wiley & Sons, New York, 1965.
[4] O. Hamsici, A. Martinez, Bayes optimality in linear discriminant analysis, IEEE Trans. Pattern Anal. Mach. Intell. 30 (4) (2008) 647–657.
[5] N.A. Campbell, Canonical variate analysis—a general formulation, Aust. J. Stat. 26 (1984) 86–96.
[6] T. Hastie, R. Tibshirani, B. Andreas, Flexible discriminant and mixture models, in: Statistics and Neural Networks: Advances at the Interface, 1999, pp. 1–23.
[7] N. Kumar, A. Andreou, Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition, Speech Commun. 26 (4) (1998) 283–297.
[8] M. Zhu, T. Hastie, Feature extraction for nonparametric discriminant analysis, J. Comput. Graph. Stat. 12 (1) (2003) 101–120.
[9] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (10) (2000) 2385–2404.
[10] S. Mika, G. Rätsch, K.-R. Müller, A mathematical programming approach to the kernel Fisher algorithm, in: NIPS 13, 2000, pp. 591–597.
[11] M. Loog, R. Duin, R. Haeb-Umbach, Multiclass linear dimension reduction by weighted pairwise Fisher criteria, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 762–766.
[12] J. Lu, K. Plataniotis, A. Venetsanopoulos, Face recognition using LDA-based algorithms, IEEE Trans. Neural Netw. 14 (1) (2003) 195–200.
[13] M. Zhu, A. Martinez, Subclass discriminant analysis, IEEE Trans. Pattern Anal. Mach. Intell. 28 (8) (2006) 1274–1286.
[14] S. Zhang, T. Sim, Discriminant subspace analysis: a Fukunaga–Koontz approach, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1732–1745.
[15] F. De La Torre, T. Kanade, Multimodal oriented discriminant analysis, in: Proceedings of ICML, 2005, pp. 177–184.
[16] S. Kullback, Information Theory and Statistics, Dover, New York, 1997.
[17] D. Tao, X. Li, X. Wu, S. Maybank, General averaged divergence analysis, in: IEEE Proceedings of 7th ICDM, 2007, pp. 302–311.
[18] K.T. Abou-Moustafa, F. De La Torre, F.P. Ferrie, Pareto discriminant analysis, in: IEEE Proceedings of 23rd CVPR, 2010, pp. 3602–3609.
[19] R. Chen, Solution of MINIMAX problems using equivalent differentiable functions, Comput. Math. Appl. 12 (1985) 1165–1169.
[20] Y. Sawaragi, H. Nakayam, T. Tanino (Eds.), Theory of Multiobjective Optimization, Academic Press, Orlando, FL, 1985.
[21] C. Hillermeier, Nonlinear Multiobjective Optimization, Birkhäuser Verlag, Heidelberg, Germany, 2001.
[22] M. Ehrgott, Multicriteria Optimization, Springer, New York, 2005.
[23] T. Hastie, R. Tibshirani, Discriminant analysis by Gaussian mixtures, J. R. Stat. Soc. Ser. B Methodol. 58 (1) (1996) 155–176.
[24] J. Tou, R. Heydorn, Some approaches to optimum feature selection, Comput. Inf. Sci. II (1967) 57–89.
[25] G. Saon, M. Padmanabhan, R. Gopinath, S. Chen, Maximum likelihood discriminant feature spaces, in: ICASSP, 2000.
[26] M. Loog, R.P.W. Duin, Linear dimensionality reduction via a heteroscedastic extension of LDA: the Chernoff criterion, IEEE Trans. Pattern Anal. Mach. Intell. 26 (6) (2004) 732–739.
[27] Q. Gao, J. Liu, H. Zhang, J. Hou, X. Yang, Enhanced Fisher discriminant criterion for image recognition, Pattern Recognit. 45 (10) (2012) 3717–3724.
[28] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.
[29] R. Lotlikar, R. Kothari, Fractional-step dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 22 (6) (2000) 623–627.
[30] H. Yu, J. Yang, A direct LDA algorithm for high-dimensional data—with application to face recognition, Pattern Recognit. 34 (2001) 2067–2070.
[31] E.K. Tang, P.N. Suganthan, X. Yao, A.K. Qin, Linear dimensionality reduction using relevance weighted LDA, Pattern Recognit. 38 (4) (2005) 485–493.
[32] W. Bian, D. Tao, Max-min distance analysis by using sequential SDP relaxation for dimension reduction, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 1037–1050.
[33] H.P. Decell, J.A. Quirein, An iterative approach to the feature selection problem, in: LARS Symposia, 1973, pp. 3B-1–9.
[34] K.B. Petersen, M.S. Pedersen, The Matrix Cookbook, November 2008.
[35] C. Ding, T. Li, Adaptive dimension reduction using discriminant analysis and K-means clustering, in: Proceedings of 24th ICML, 2007.
[36] S. Dasgupta, Experiments with random projection, in: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2000, pp. 143–151.
[37] S. Laue, A hybrid algorithm for convex semidefinite optimization, in: Proceedings of 29th ICML, 2012.
[38] S. Raudys, A. Raudys, Pairwise costs in multiclass perceptrons, IEEE Trans. Pattern Anal. Mach. Intell. 32 (7) (2010) 1324–1328.
[39] S.X. Yu, J. Shi, Multiclass spectral clustering, in: IEEE Proceedings of ICCV, 2003, pp. 313–319.
[40] D. Newman, S. Hettich, C. Blake, C. Merz, UCI Repository of Machine Learning Databases ⟨www.ics.uci.edu/~mlearn/MLRepository.html⟩, 1998.
[41] M. Zhu, A. Martinez, Pruning noisy bases in discriminant analysis, IEEE Trans. Neural Netw. 19 (1) (2008) 148–157.
[42] J.J. Hull, A Database for Handwritten Text Recognition Research ⟨www.gaussianprocess.org/gpml/data/⟩, 1994.
[43] Y. LeCun, The MNIST Database of Handwritten Digits ⟨http://yann.lecun.com/exdb/mnist/⟩, 1998.
[44] T. Sim, T. Kanade, Combining models and exemplars for face recognition: an illuminating example, in: CVPR Workshop on Models vs. Exemplars in Computer Vision, 2001.
[45] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.
[46] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.
[47] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning a Mahalanobis metric from equivalence constraints, J. Mach. Learn. Res. 6 (2005) 937–965.
[48] T. Hastie, R. Tibshirani, J. Friedman (Eds.), Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer, New York, NY, 2001.
[49] A.M. Martinez, A. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233.

Karim T. Abou-Moustafa received his B.Eng. in Computer Engineering from the Arab Academy for Science and Technology, Alexandria, Egypt, in 1999; M.Sc. in Computer Science from Concordia University, Montreal, QC, Canada, in 2004; and the Ph.D. in Electrical & Computer Engineering from McGill University, Montreal, QC, Canada, in 2012. He was a Research Asst. in the Centre of Pattern Recognition and Machine Intelligence, Concordia University, from 2001 to 2004, a Research Asst. in Laboratoire d'Information et System Adaptive, Universite de Montreal, from 2004 to 2005; a Visiting Scholar in the Robotics Institute, Carnegie Mellon University (CMU), Pittsburgh, PA, in 2009; and a Postdoctoral Fellow in the Robotics Institute, CMU, from 2011 to 2012. He was awarded the PRECARN Student Scholarship, in 2004; the FQRNT–REPARTI Scholarship for International Training, in 2009; and the FQRNT Postdoctoral Fellowship 2011–2013. Since 2012, Dr. Abou-Moustafa is a Postdoctoral Research Fellow in the Dept. of Computing Science, University of Alberta, AB, Canada. His main research interest is the design of computationally efficient learning algorithms.

Fernando De la Torre received his B.Sc. degree in Telecommunications, M.Sc. and Ph.D. degrees in Electronic Engineering, respectively, in 1994, 1996 and 2002, from La Salle School of Engineering in Ramon Llull University, Barcelona, Spain, respectively. In 1997 and 2000 he became an Assistant and an Associate Professor in the Department of Communications and Signal Theory in La Salle School of Engineering. In 2003 he joined the Robotics Institute at Carnegie Mellon University and he is currently an Associate Research Professor. His research interests are in the fields of Computer Vision and Machine Learning. Currently, he is directing the component analysis lab (http://ca.cs.cmu.edu) and the human sensing lab (http://humansensing.cs.cmu.edu). Dr. De la Torre has co-organized the first workshop on component analysis methods for modeling, classification and clustering problems in computer vision in conjunction with CVPR'07 and the workshop on human sensing from video in conjunction with CVPR'06. He has also given several tutorials at international conferences on the use and extensions of component analysis methods.

Frank P. Ferrie received the B. Eng., M. Eng. and Ph.D. degrees in Electrical Engineering from McGill University in 1978, 1980, and 1986, respectively. He is currently a Professor in the Department of Electrical and Computer Engineering at McGill and co-director of the REPARTI research network in distributed environments. From 1998 to 2001 he served as a Director of the McGill Centre for Intelligent Machines, from 2002–2004 as a Director of the Québec Research Network in Distributed Reality, and from 2005–2006 as an Associate Dean, Research and Graduate Studies in the Faculty of Engineering. His research interests are in the area of Computer Vision, primarily in the areas of Two and Three-dimensional Shape Analysis, Active Perception, dynamic Scene Analysis and Machine Vision. He has published extensively in these areas, and is best known for his work in active vision and environmental modeling. In recognition of these contributions he was awarded the 2011 Award for Research Excellence and Service to the Research Community by the Canadian Image Processing and Pattern Recognition Society. He is a member of the IEEE, the IEEE Computer Society, and registered in the Association of Professional Engineers of Ontario.