Novel Fisher discriminant classifiers

Alessandro Rozza*, Gabriele Lombardi, Elena Casiraghi, Paola Campadelli

Dipartimento di Scienze dell’Informazione (DSI), Università degli Studi di Milano, Via Comelico 39/41, I-20135 Milan, Italy

ARTICLE INFO

Article history:
Received 4 May 2011
Received in revised form 6 March 2012
Accepted 13 March 2012

Keywords:
Fisher subspace
Supervised learning
Discriminant techniques
Small sample size problem

ABSTRACT

At the present, several applications need to classify high dimensional points belonging to highly unbalanced classes. Unfortunately, when the training set cardinality is small compared to the data dimensionality (“small sample size” problem) the classification performance of several well-known classifiers strongly decreases. Similarly, the classification accuracy of several discriminative methods decreases when non-linearly separable, and unbalanced, classes are treated. In this paper we firstly survey state of the art methods that employ improved versions of Linear Discriminant Analysis (LDA) to deal with the above mentioned problems; secondly, we propose a family of classifiers based on the Fisher subspace estimation, which efficiently deal with the small sample size problem, non-linearly separable classes, and unbalanced classes. The promising results obtained by the proposed techniques on benchmark datasets and the comparison with state of the art predictors show the efficacy of the proposed techniques.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

In 1957 the first binary linear classifier, the linear “perceptron”, was proposed by Rosenblatt [1]. While the development of this algorithm initially generated some enthusiasm, partly because of its apparent relation to biological mechanisms, the later discovery of its inadequacy to discriminate non-linearly separable classes caused such model to be abandoned and raised an increasing interest toward the development of more complex classifiers. For this reason, supervised linear and non-linear artificial neural networks (ANN) were subsequently proposed, and gained a lot of attention in the 1980s.

After these pioneering works, several more robust and effective classifiers were proposed, such as nearest neighbor classifiers (NN [2]), probabilistic classifiers [3] (e.g. Bayesian classifiers [4]), nested dichotomies [5], random forests (RF [6]), classifiers employing linear discriminant analysis (LDA [7]) or Fisher linear discriminant analysis (FLDA [8]), the tensor voting framework (TVF [9,10]), and support vector machines (SVM [11–13]). Unfortunately, although a great deal of research work has been devoted to the development and improvement of different learning machines, most of them are affected by practical and theoretical limits that strongly reduce their performance when the following situations occur.

At first, several real life classification problems have at their disposal a limited number of training data [14]; in this case, classifiers with sufficient flexibility but also sufficient robustness against overfitting are needed to avoid a strong decrease of the classification accuracy during the testing phase. On the other hand, when the data are encoded in a high dimensional space, many techniques cannot be applied due to their high time and space complexity. Moreover, when the “small sample size problem” [15] occurs, that is when the dimensionality of the feature space is higher than the number of available training data, the underlying mathematical formulations of several learning algorithms could yield poor performance since the amount of training data is not enough to compute reliable estimates of the employed mathematical entities (see Section 3 for a more detailed explanation). Besides, a related problem is overfitting, which could happen when non-linearly separable classes must be discriminated, or when the number of available samples is too small compared to the space dimensionality and the model is complex enough.

Other problems affecting classification performance could be due to unbalanced training sets, that is training sets where the number of examples per class differs greatly across the classes, such as those processed by medical and biomedical applications [16,17]. For all the reasons cited above, classifiers that achieve high classification accuracy, while maintaining a low computational complexity, when dealing with non-linearly separable, unbalanced, and high dimensional datasets characterized by a small sample size are advisable and are nowadays an important subject of research.

In this work, we firstly survey interesting state of the art methods employing modified versions of LDA to cope with the above mentioned problems; secondly, we describe our research work aimed at the same purpose. More precisely, we propose a family of binary classifiers, based on the projection of the data on
the Fisher subspace estimated on the training points. We have already reported a description of these classifiers in [17–19]; nevertheless, in this paper we describe in detail the basic theories of our classifiers, and we exhaustively test them on benchmark datasets to show that they can be successfully combined to accomplish multiclass classification.

This paper is organized as follows: in Section 2 the basic theory of both the Fisher subspace and \( LDA \) are summarized; in Section 3 the most important improvements of \( LDA \) are recalled, highlighting their advantages and their drawbacks; in Section 4 our base classifier, called IPCAC, is summarized; in Section 5 an improved version, called T-IPCAC, and its online version, called O-IPCAC, are described; Section 6 contains the kernel versions of IPCAC and T-IPCAC, which are, respectively, called K-IPCAC and K-TIPCAC; in Section 7, we firstly describe how to combine our binary classifiers to perform multiclass classification (Section 7.1), and then we test the developed ensemble techniques on benchmark datasets characterized by high dimensionality and affected by the small sample size problem (Sections 7.2 and 7.3). The achieved results are critically analyzed by comparing them with those obtained by state of the art methods related to our work; finally, conclusions and future works are summarized in Section 8.

2. Theoretical background

In this section we focus on the two-class classification problem and we summarize the basic theory of linear discriminant analysis (\( LDA \) [7]). \( LDA \) is strongly related to Fisher linear discriminant analysis (FLDA) [8], since it employs a slightly different approach that assumes normally distributed classes and homoscedastic datasets (that is datasets characterized by identical within-class covariance matrices).

To recall the binary \( LDA \) method, we consider a set of clustered points \( \mathcal{P} = \mathcal{P}_A \cup \mathcal{P}_B \) drawn from \( \mathcal{N}^D \), where \( A \) and \( B \) are two classes to be discriminated, and \( \mathcal{P}_A \) and \( \mathcal{P}_B \) are two clusters containing \( N_A \) and \( N_B \) points belonging to classes \( A \) and \( B \), respectively. In this case \( LDA \) firstly projects the points on the one-dimensional Fisher subspace \( Fs_p \), defined by the given point set \( \mathcal{P} \); \( Fs_p \) is defined as the subspace, spanned by a vector \( \mathbf{w} \in \mathcal{N}^D \), that maximizes Fisher’s criterion, \( J(w) \), that is a function whose maximization produces a large variance between the projected class means (which for the two-class case corresponds to a large separation between the projected class means) while simultaneously minimizing the variance within each projected class. More precisely, \( J(w) \) is formulated in the two-class case as

\[
J(w) = \frac{\text{between-class variance of } \mathcal{P} \text{ proj on } w}{\text{within-class variance of } \mathcal{P} \text{ proj on } w} = \frac{\mathbf{w}^T \Sigma_{\text{Bet}} \mathbf{w}}{\mathbf{w}^T \Sigma_{\text{Wt}} \mathbf{w}}
\]

where \( \Sigma_{\text{Bet}} \) and \( \Sigma_{\text{Wt}} \), which are, respectively, the between-class covariance matrix and the total within-class covariance matrix, are defined as

\[
\Sigma_{\text{Bet}} = (\mu_A - \mu_B)(\mu_A - \mu_B)^T, \quad \Sigma_{\text{Wt}} = \frac{1}{N_A} \sum_{p \in \mathcal{P}_A} (p - \mu_A)(p - \mu_A)^T + \frac{1}{N_B} \sum_{p \in \mathcal{P}_B} (p - \mu_B)(p - \mu_B)^T,
\]

where \( \mu_A \) and \( \mu_B \) are the estimated class means:

\[
\mu_A = \frac{1}{N_A} \sum_{p \in \mathcal{P}_A} p, \quad \mu_B = \frac{1}{N_B} \sum_{p \in \mathcal{P}_B} p
\]

Differentiating Eq. (1) with respect to \( \mathbf{w} \), we find that \( J(w) \) is maximal when

\[
(\mathbf{w}^T \Sigma_{\text{Bet}} \mathbf{w}) \Sigma_{\text{Wt}} \mathbf{w} = (\mathbf{w}^T \Sigma_{\text{Wt}} \mathbf{w}) \Sigma_{\text{Bet}} \mathbf{w}.
\]

Considering that \( \Sigma_{\text{Bet}} \mathbf{w} \) is a vector oriented along \( (\mu_A - \mu_B) \), and that both \( (\mathbf{w}^T \Sigma_{\text{Bet}} \mathbf{w}) \) and \( (\mathbf{w}^T \Sigma_{\text{Wt}} \mathbf{w}) \) can be dropped since the magnitude of \( \mathbf{w} \) does not affect the quality of the estimation, we obtain \( \mathbf{w} \propto (\mu_A - \mu_B) \).

The resulting vector, which could also be computed by finding the eigenvector of \( \Sigma_{\text{Wt}} \Sigma_{\text{Bet}} \) corresponding to the largest eigenvalue, is used to project data points, belonging to the two classes \( A \subseteq \mathcal{N}^D \) and \( B \subseteq \mathcal{N}^D \), onto a one-dimensional subspace.

To perform classification, the training points are projected on \( \mathbf{w} \), and a threshold \( \gamma \) that best separates the data is estimated. An unknown point \( \mathbf{p} \) is then classified as belonging to class \( A \) if \( \mathbf{w} \cdot \mathbf{p} \leq \gamma \) (where \( \cdot \) is the dot product operator), to class \( B \) otherwise.

Note that, when the assumptions of \( LDA \) are satisfied, that is the classes are normally distributed and the covariance matrices within each class are identical (homoscedasticity assumption) and have full rank, the projection vector computed by FLDA is equivalent to that obtained by \( LDA \), and it is the Bayes optimal solution, commonly called ”optimal discriminant projection vector”. However, since \( LDA \) is limited by the implicit assumption of homoscedastic data, several authors have proposed different extensions of \( LDA \) to also consider the discriminative information carried by covariance differences. A good survey has been reported in [20], where the authors have also proposed a simple and effective \( LDA \) extension; more precisely, to take into account the discriminative information that is present in the difference between the within-class scatter matrices, the authors modify the Fisher criterion by substituting the between-class scatter matrix with a “direct distance matrix” [21], which is based on the Chernoff distance [22]. This measure of affinity of two densities is particularly suited for extending \( LDA \) to heteroscedastic data since it considers class mean differences as well as covariance differences. This method and its modified version [23] have achieved promising results on real-world datasets.

3. Improved versions of \( LDA \)

At the state of the art, many techniques based on \( LDA \) have been proposed to deal with high dimensional data and with the small sample size problem. We recall that this problem occurs when the number of training samples is lower than the space dimensionality, hereby producing singular within-class scatter matrices, which make it difficult to compute the set of projection vectors spanning the discriminant subspace. Indeed, as noted by Friedman in [24], problems also occur when the class sample sizes \( n_A, n_B \) are small compared to the dimension of the measurement space \( D \); in this case, the within-class scatter matrices become highly variable. Moreover, when \( n_k < D(k = A,B) \) not all of their parameters are even identifiable. As further explained by Friedman, the effect this has on discriminant analysis can be seen by noting that the equations employed by \( LDA \) based techniques contain the inverse of the total within-class scatter matrix, and are therefore heavily weighted by its smallest eigenvalues and the directions associated with their eigenvectors. However, as noted by the author, the sample covariance matrices produce biased estimates of the eigenvalues (the largest ones are biased high and the smallest ones are biased toward values that are too low), and this phenomenon becomes more pronounced as the sample size decreases; the net effect of this biasing phenomenon on discriminant analysis is to (sometimes dramatically) exaggerate the importance associated with the low-variance subspace spanned by the eigenvectors corresponding to the smallest eigenvalues. Therefore, most of the variance incurred in estimating projection subspace of \( LDA \) is associated with directions of low sample variance in the measurement space. In this section
we briefly overview some of the most important techniques that try to cope with these problems.

The first interesting attempts to tackle the small sample size problem, which are presented in [25–27], applied linear algebra techniques to solve the numerical problems due to the singularity of the sample within-class covariance matrix. As an example, in [25] the authors employ the pseudo-inverse of the scatter matrix to compute the Fisher subspace; alternatively, in [26,27] the authors recommend the addition of a small perturbation to the within-class scatter matrix so that it becomes non-singular. We note that a similar, and more recent, technique is presented in [28], where the authors employ a regularized version of LDA that achieves promising results on microarray data.

Although interesting results have been shown in the above mentioned works, the most promising and recent techniques employ a subspace approach. Among them, one of the most notable methods is a two-stage PCA+LDA method [29,30]; this technique firstly applies principal component analysis (PCA [31]), to evaluate the principal components used to project the points into a subspace where the sample within-class scatter matrix is not degenerate, and then employs the LDA algorithm in the PCA space. The drawback of this method is that the dimensionality reduction step might discard important discriminative information, thus decreasing the classification performance. As an example, consider two classes with the shape of two parallel “pancakes” in $\mathbb{R}^D$, i.e. two Gaussian distributions isotropic in $D–1$ directions and narrow in the last direction (see Fig. 1), so that a hyperplane orthogonal to the last direction separates the two. To perform dimensionality reduction under such circumstances, PCA would analyze the whole dataset structure (neglecting the separation among classes, as well as their structures) to keep the directions of its maximum elongation; this procedure would therefore discard the last direction, which is the most important to separate the two classes. As a consequence, the result of the dimensionality reduction process would be a mixture of two completely overlapped Gaussian distributions.

To avoid this information loss, Liu et al. [32] proposed a different approach exploiting a modified Fisher’s criterion. More precisely, in cases of non-singular $\Sigma_W$, the authors compute the optimal discriminant projection vector by selecting the eigenvectors corresponding to the set of largest eigenvalues of $\Sigma_W^{-1} \Sigma_B$, where $\Sigma_B = \Sigma_W + \Sigma_B$ is the total covariance matrix; on the other hand, when $\Sigma_W$ is singular the authors estimate the Fisher subspace from the points projected in the null space of $\Sigma_W$, that is the subspace $R_{null} \subset \mathbb{R}^D$ such that $R_{null} = \text{Span}(\{z \in \mathbb{R}^D \mid \Sigma_W z = 0, i = 1, \ldots, D-r\})$ where $r$ is the rank of $\Sigma_W$. Note that the existence of this space is guaranteed by the fact that $r$ is lower than the space dimensionality $D$.

From the linear algebra it can be proved that for each $w \in R_{null}$ then $w^T \Sigma_W w = 0$ which yields $w^T \Sigma_B w = w^T \Sigma_W w$. In this case, the authors note that the modified Fisher’s criterion becomes equal to 1, that is its maximum, for each $w \in R_{null}$ such that $w^T \Sigma_B w \neq 0$. However, a projection vector $w \in R_{null}$ satisfying the above mentioned conditions cannot guarantee the maximum class separability, unless $w^T \Sigma_B w$ is further maximized. Therefore, Chen et al. [15] propose to choose as projection vectors those vectors in the null space that maximize the between-class scatter of the training samples projected on $R_{null}$. This procedure is based on the authors’ observation that the null space of $\Sigma_W$ carries the most of the discriminative information. Intuitively, choosing projection vectors such that $w^T \Sigma_B w = 0$ while $w^T \Sigma_W w \neq 0$ means that, when projected onto $w$, the within-class separation is eliminated while the between-class scatter is not. Obviously, this method can increase the discriminative quality of the estimated Fisher subspace, but it does not consider any information outside the null space of $\Sigma_W$; furthermore, it has some computational limitations in managing large sample covariance matrices.

To overcome these limitations, in [33] the authors proposed a “unified” PCA+LDA algorithm (D-LDA) which works on high dimensional data by simultaneously performing the two steps. The key idea of this algorithm is to discard the null space of $\Sigma_B$, which does not contain useful discriminative information, maintaining the information contained in the null space of $\Sigma_W$ that, as suggested by Chen, is the most discriminative part. To this aim, the authors proceed as follows: firstly, diagonalize $\Sigma_B$, to find a matrix $V$ such that $V^T \Sigma_B V = \Lambda$, $V^T = I$ and $\Lambda$ is a diagonal matrix sorted in decreasing order; secondly, diagonalize $Y^T \Sigma_W Y$, where $Y$ is composed of the first columns of $V$, and discard the largest eigenvalue/eigenvector pairs to keep the information in the null space of $\Sigma_W$. Although this approach reverses the traditional procedure order, when $\Sigma_W$ is not singular it is equivalent to PCA+LDA [30]; otherwise, when $\Sigma_W$ is singular the reversal in order makes a drastic difference.

Although promising results have been obtained by this method, Lu et al. [34] noted that its performance deteriorates rapidly when the small sample size problem becomes severe, since the insufficient number of high dimensional training examples makes it difficult to compute a reliable estimate of the null eigenvalues of the within-class covariance matrix, and hereby its null space. Indeed, in this case both bias and variance affect the estimation of the within-class scatter matrix. Firstly, the estimated $\Sigma_W$ produces biased estimated eigenvalues, so that the largest ones are biased high and the smallest ones are biased toward too low values. Secondly, the estimate of the null space of $\Sigma_W$ can be highly unstable, giving rise to high variance. Based on these considerations, Lu et al. proposed a new LDA methodology (R-LDA [34,35] employing a regularized Fisher’s discriminant criterion, that is

$$J(W) = \frac{W^T \Sigma_B W}{(W^T \Sigma_B W) + W^T \Sigma_W W}$$

where $0 \leq v \leq 1$, which is equivalent to both D-LDA and LDA under particular configurations of the regularization parameter $v$. This method is inspired by the regularized quadratic discriminant.
analysis (FDA) proposed by Friedman [24]. More precisely, in his paper Friedman firstly recalls the quadratic discriminant function, which is used when dealing with heteroscedastic data, and then introduces a regularization term to cope with the highly ill-posed covariance matrices that are estimated when the number of samples per class is too small. Friedman demonstrated that the regularization successfully decreases the larger eigenvalues and increases the smaller ones, thereby counteracting the biasing; moreover, this technique stabilizes the smallest eigenvalues. Indeed, the results presented in [34,35] showed that the technique inspired by Friedman’s approach is more robust against the small sample size problem than the D-LDA technique.

We note that, although promising results have been obtained by the above mentioned techniques, none of them consider the case where the number of training examples is comparable to the space dimensionality \( N \gg D \). In such circumstances the sample within-class covariance matrix is often non-singular, since it might be \( N > D \), but the sample covariance matrix is not a consistent estimator (when \( N \to D, N \to +\infty, \) and \( D \to +\infty \)) of the population covariance matrix, as proved in [36]. Under this setting, the estimated null space might not contain useful information, thus causing a drastic performance decrease.

We further note that all these techniques are based on linear classifiers; therefore, their performance significantly decreases with non-linearly separable classes. To handle this problem, a key idea would be to employ the kernel principal component analysis (KPCA [37]) and the kernel Fisher discriminant (KFD [38,39]), which are the kernel versions of PCA and LDA, respectively, and are related to the kernel version of our base linear classifier (see Sections 4 and 6).

More precisely, KPCA exploits the “kernel trick” to project the points in a higher dimensional feature space where they are supposed to be linearly separable. Note that this dimensionality reduction technique has been applied as the first step to simplify difficult multiclass classification problems characterized by high dimensional data, such as face recognition [40]. Exploiting the same idea, KFD performs LDA on a high dimensional feature space implicitly defined by a kernel; more specifically, in [39] the authors demonstrate that the kernel Fisher discriminant method is equivalent to KPCA plus LDA. Based on this result, the authors proposed a more transparent KFD algorithm based on a first step of KPCA followed by LDA for a second feature extraction in the KPCA-transformed space.

It is important to notice that KFD has been successfully applied to real life applications such as face recognition [41–44], where the small sample size problem often occurs since faces are generally encoded through high dimensional vectors and only few samples are typically available for each person (class). To tackle face recognition problems, in [42] the authors directly employ KFD on 2D face images, while a more complicated technique is described in [41] to recognize persons’ identity from videos. In this paper, the authors describe a multi-view dynamic face model designed to extract the shape-and-pose-free facial texture patterns for accurate across-view registration; subsequently KFD is employed to compute the non-linear discriminating basis vectors, so that identity surfaces of face classes are constructed in this discriminating feature space. Finally, face recognition is performed by matching a “face trajectory”, tracked from an input video and projected on the KFD subspace, with a set of model trajectories synthesized on the identity surfaces.

Another KFD-based face recognition approach is the one presented in [44], where the authors define the KFD-Isomap method, as an extended version of Ext-Isomap [45,46]. More precisely, Ext-Isomap exploits the geodesic distance, computed as described in [47], to represent each data point by a feature vector of its geodesic distance to any other training point. Subsequently, LDA is applied on the feature vectors to find an optimal projection direction for classification. Classification of unknown feature vectors is then performed by projecting them on the estimated Fisher subspace and then employing the nearest neighbor classifier. KFD-Isomap [44] employs a similar procedure, and substitutes LDA with KFD, to non-linearly map the feature vectors into a high dimensional space where LDA is then employed to find the projection directions maximizing the distances between cluster centers.

Although the above mentioned kernel methods achieve the best face recognition results, as it generally happens for most kernel-based classification methods, their classification performance strongly depends on the choice of the kernel, and they often suffer from overfitting problems. To specifically cope with overfitting and the small sample size problem, in [48] the authors extend both the LDA-GSVD algorithm [49,50] and its kernel versions, called KDA-GSVD [51] and mGSVD-KDA [52]. More precisely, the LDA-GSVD algorithm [49,50], which has been designed to cope with the small sample size problem, employs the generalized singular value decomposition (GSVD [53]) to solve a generalized eigenvalue problem. The application of GSVD to LDA not only provides a framework for finding the feature vectors with high recognition accuracy, but also relaxes the non-singularity requirement. However, this algorithm encounters excessive computational problems when the samples have large dimensionality, and, being a linear algorithm, it still cannot handle non-linear separable data. For this reason, in [51,52] two kernel version of LDA-GSVD are proposed. In [51] the authors propose the KDA-GSVD approach that applies GSVD to solve the generalized eigenvalue problem which is formulated in the feature space defined by a non-linear mapping through kernel functions. In [52] a similar solution is proposed, called mGSVD-KDA; more precisely, the original input samples are non-linearly mapped into a higher dimensional space where the pattern distribution is linearized, and a modified GSVD scheme is used to extract features in that space. The modified GSVD has been designed to overcome the computational complexity problem associated with the high dimensional patterns. Though promising results have been obtained by exploiting the cited GSVD based algorithms, as mentioned above, they still suffer from overfitting problems; therefore, in [48] the authors propose an orthogonalization of the basis of the discriminant subspace derived from both the LDA-GSVD and the mGSVD-KDA algorithms. When tested on face recognition, this algorithm achieves promising results, which seem comparable to those reported in [44].

In this paper we present a family of binary (linear and non-linear) classifiers, and their combination to perform multiclass classification. These classifiers are designed to cope with the small sample size problem, and are able to efficiently classify high dimensional data and non-linearly separable classes, by exploiting the Fisher subspace estimated on the training data through an efficient approach.

### 4. Isotropic PCA-based classifier

In this section we describe a linear binary classifier, called isotropic principal component analysis classifier (IPCA), which deals with data points drawn from a mixture of two Gaussians (MoG), one multivariate Gaussian distribution per class. IPA can be considered as an evolution of the linear discriminant analysis (LDA), since it estimates the Fisher subspace \( (FS) \) by employing a novel and efficient approach, which is based on the theoretical results reported by Brubaker and Vempala in [54].

More precisely the authors proved that, considering a MoG distribution in \( \mathbb{R}^D \) having the overall mean \( \mu = 0 \) and overall covariance matrix \( \Sigma = \sigma^2 I \) (where \( \sigma \) is the standard deviation and \( I \) is the identity matrix), if \( \mu_i \in \mathbb{R}^D \) is the mean vector of the \( i \)-th
Gaussian distribution in the mixture, then the subspace spanned by all the \( \mu_i (1 \leq i \leq C) \) is the Fisher subspace.

Exploiting this result, in a two-class classification problem, supposing that the classes are \( A \) and \( B \), and that the points in the training set \( \mathcal{P} = \mathcal{P}_A \cup \mathcal{P}_B = \{ \mathbf{p}_i \}_{i=1}^N \) (where \( \mathcal{P}_A \) and \( \mathcal{P}_B \) are the subsets containing points belonging to classes \( A \) and \( B \), respectively) are sampled from two Gaussians, we represent the estimate of the one-dimensional \( \mathbf{F}_s \) by the unit vector \( \mathbf{f} \), computed as follows:

\[
\mathbf{f} = \frac{\mu_A - \mu_B}{\| \mu_A - \mu_B \|} \tag{2}
\]

where \( \mu_A \in \mathbb{R}^D \) and \( \mu_B \in \mathbb{R}^D \) are the means of the training points belonging to classes \( A \) and \( B \), respectively.

Given an unknown point \( \mathbf{p} \), it is classified by projecting it on \( \mathbf{F}_s \), and then thresholding the obtained value \( \text{proj}(\mathbf{p}) = \mathbf{f} \cdot \mathbf{p} \).

In practice, the probability distribution related to several classification tasks employing real data is not mean-centered, and its random variables are often correlated. For this reason, we preprocess the data by applying a linear whitening transformation.\(^2\) To this aim, considering the set of \( N \) training points \( \mathcal{P} = \mathcal{P}_A \cup \mathcal{P}_B = \{ \mathbf{p}_i \}_{i=1}^N \), and \( \mu = \sum \mathbf{p}_i / N \) is the sample mean, the whitening matrix \( \mathbf{W} \) is estimated by employing the eigenvalues and eigenvectors of the sample covariance matrix \( \Sigma = \sum\{(\mathbf{p}_i - \mu)(\mathbf{p}_i - \mu)/N \} \).

\[
\Sigma = \mathbf{X} \Lambda \mathbf{X}^T \quad \mathbf{W} = \mathbf{X} \Lambda^{-1/2} \mathbf{X}^T \tag{3}
\]

Note that \( \Lambda^{-1/2} \) can be computed by substituting the non-zero diagonal elements \( \lambda_i \) of \( \Lambda \) with the values \( \lambda_i^{1/2} \).

The whitened training data, obtained through \( \mathbf{W} \) and \( \mathbf{X} \), are employed to compute the class means \( \mu_A \) and \( \mu_B \); \( \mathbf{f} \) is then computed by means of Eq. (2). Therefore, given a new point \( \mathbf{p} \), it is projected on \( \mathbf{f} \) as follows:

\[
\text{proj}(\mathbf{p}) = \mathbf{f}^T (\mathbf{W}(\mathbf{p} - \mu) = \mathbf{f} \cdot (\mathbf{p} - \mu) \tag{4}
\]

By thresholding its projection on \( \mathbf{f} \), in [18] the threshold value was selected by considering the set of projected training points \( \mathcal{T}_P = \{ \mathbf{w}^T (\mathbf{p} - \mu) \mid \mathbf{p} \in \mathcal{P} \} \), and computing the average of those projected training points that maximize the function \( \text{Score}(\mathcal{T}) \), which computes the number of correctly classified points when \( \gamma \) is used as threshold, that is

\[
\gamma = \left\langle \arg \max_{\mathbf{p} \in \mathcal{T}_P} \text{Score}(\mathcal{T}) \right\rangle \tag{5}
\]

where \( \langle \cdot \rangle \) is the mean operator applied to the set of threshold values that give the same best value of the \( \text{Score}(\cdot) \) function.

However, since this threshold selection method produced poor results in case of unbalanced training data, in this paper we employ a different threshold selection method, which is based on the fact that, by hypothesis, the two-class distributions are Gaussians with different within-class covariances. Since the class projections on the Fisher subspace are still distributed as Gaussian density functions with non-equal variances, we have chosen to set the value of \( \gamma \) so that it corresponds to the point having the same Mahalanobis distance \( \| \xi \| \) from the projections of the mean vectors on the Fisher subspace. More precisely, considering the projected mean vectors, that are \( \mathcal{P}_A = \mathbf{W} \cdot (\mu_A - \mu) \) and \( \mathcal{P}_B = \mathbf{W} \cdot (\mu_B - \mu) \), we impose the constraint \( \mathcal{P}_A = \xi \sigma_A = \mathcal{P}_B = \xi \sigma_B \),

where \( \sigma_A \) and \( \sigma_B \) are the standard deviations of the whitened points projected on \( \mathbf{w} \).

Defining \( \gamma = \mathcal{P}_A + \xi \sigma_A \) we obtain

\[
\gamma = \mathcal{P}_A + \xi \sigma_A \tag{6}
\]

\( \text{IPCA} \) is very simple, and it has a low computational cost both in the training and in the classification phase. Moreover, denoting with \( \mathbf{D} \) the feature space dimensionality, the space complexity required to store an \( \text{IPCA} \) model is \( 2D + 1 \): real values are needed to store the estimated expectation\(^4\) \( \mu, \sigma_D \) real values record the weight vector \( \mathbf{w} \), and one value stores the computed threshold value.

Considering the time complexity, this method is linear in the classification phase, while the time spent for training is dominated by the SVD computation, which requires \( O(\min(N^2 D, N D^2)) \) [55].

5. \( T\text{-IPCA} \) and \( O\text{-IPCA} \)

Despite the efficiency and the effectiveness of \( \text{IPCA} \), shown by the experiments proposed in [18], its performance is highly affected when the cardinality of the training dataset \( N \) is lower than, or comparable to, the space dimensionality \( D \), that is \( \alpha = D/N \geq 1 \). In this section we firstly recall the problems that might arise when these situation occur, and then we propose \( T\text{-IPCA} \) (truncated \( \text{IPCA} \)) and its online version (\( O\text{-IPCA} \)), which are two improved versions of \( \text{IPCA} \).

Given the matrix \( \mathbf{P} \in \mathbb{R}^{D \times N} \), representing a training dataset \( \mathcal{P} = \mathcal{P}_A \cup \mathcal{P}_B \), where \( |\mathcal{P}| = N = N_A + N_B \), let \( \alpha \) be the ratio \( D/N \).

When \( \alpha > 1 \) the small sample size causes the sample covariance matrix to be singular, and \( \text{IPCA} \) cannot apply the whitening step in the \( D \)-dimensional space. On the other hand, when \( \alpha \approx 1 \), the performance of \( \text{IPCA} \) deteriorates dramatically since the sample covariance matrix

\[
\Sigma = \frac{1}{N-1} \mathbf{P} \mathbf{P}^T \]

is not a consistent estimator of the population covariance matrix \( \Sigma \) (considering \( N \to +\infty \) and \( D \to +\infty \)) [36,56], thus affecting the classification performance. Besides, when \( \alpha \approx 1 \) the estimates of the smallest eigenvalues can be much smaller than the real ones, and the corresponding estimated eigenvectors are uncorrelated with the real ones.

It has to be further noted that, when the cardinality of the training set and the input space dimensionality are high, the high space complexity can cause difficulties.

To solve these problems we replaced the \( \text{IPCA} \) whitening step by a “partial whitening”, that operates only on the \( d \)-dimensional subspace \( \pi_d = \text{Span} \{ \mathbf{v}_1, \ldots, \mathbf{v}_d \} \) which captures most of the data variance,\(^4\) while maintaining unaltered the information related to the orthogonal subspace \( \left( \pi_d \right) ^\perp = \text{Span} \{ \mathbf{v}_{d+1}, \ldots, \mathbf{v}_D \} \).

To perform the “partial whitening”, we must firstly estimate \( \pi_d \) by computing the \( d \) eigenvectors corresponding to the \( d \) largest eigenvalues of the sample covariance matrix

\[
\Sigma = \frac{1}{N-1} \mathbf{P} \mathbf{P}^T \]

However, it is well known that the computation of both \( \Sigma \) and its eigendecomposition \( \Sigma = \mathbf{U} \mathbf{U}^T \) can be replaced by the computation of the truncated singular value decomposition

\[ \text{Note that considering the equivalent thresholding function } w \neq \gamma + w \delta, \text{ the space required to store } w \text{ can be avoided.} \]

\[ \text{We underline that, to avoid the problems due to the singularity of the sample covariance matrix, the value of } d \text{ must be smaller than } \min(D,N); \text{ in our experiments this value is empirically chosen.} \]
between the rank factorization $\mathbf{SP}$ of the sample matrix $\mathbf{P}$; this step computes the low-rank factorization $\mathbf{P} \approx \mathbf{U}_d \mathbf{Q}_d \mathbf{V}_d^T$, where $\mathbf{Q}_d$ is a diagonal matrix containing the $d$ largest singular values of $\mathbf{P}$, while $\mathbf{U}_d$ contains the associated left singular vectors, which span the $d$-dimensional subspace $\mathbf{SP}_d$ estimating $\mathbf{\pi}_d$. Data whitening in $\mathbf{SP}_d$ is then obtained by projecting the points in $\mathbf{P}$ on $\mathbf{SP}_d$ and by scaling them with $\mathbf{Q}_d^{-1}$, thus obtaining the matrix $\mathbf{W}_d$, containing the $d$-dimensional whitened points in $\mathbf{SP}_d$, that is: $\mathbf{W}_d = \mathbf{Q}_d^{-1} \mathbf{U}_d \mathbf{P} = \mathbf{W}_d \mathbf{P}$.

The obtained matrix $\mathbf{W}_d$ projects and whitens the points in the linear $d$-dimensional subspace $\mathbf{SP}_d$; nevertheless, this whitening step also performs dimensionality reduction, which might delete discriminative information decreasing the classification performance (as we recalled in Section 2 and showed in Fig. 1, and as it is further reported in [33]). To avoid this information loss, we add to the $d$-dimensional data, stored in $\mathbf{W}_d$, the residuals $\mathbf{R}$ of the points in $\mathbf{P}$ with respect to their projections on $\mathbf{SP}_d$, where $\mathbf{R} = \mathbf{P} - \mathbf{U}_d \mathbf{Q}_d \mathbf{V}_d^T$.

Note that, before adding $\mathbf{R}$ to the points in $\mathbf{W}_d$, we multiply them by $\mathbf{q}_d$, that is the smallest singular value found in $\mathbf{Q}_d$ and retained by the $\text{TSVD}$. This rescaling operation ensures that the $d$ singular values of $\mathbf{W}_d$ (which, thanks to the whitening process, are all equal to 1) become all equal to $q_d$; this step avoids the gap between the $d$-th and the $(d+1)$-th singular value after the residuals have been retrieved and added back. Summarizing, the data whitening in $\mathbf{\pi}_d$, the rescaling by $q_d$, and the addition of the residuals stored in the matrix $\mathbf{R}$ are computed as follows:

$$\mathbf{W}_d = \mathbf{Q}_d^{-1} \mathbf{U}_d \mathbf{P} = \mathbf{W}_d \mathbf{P} \quad \text{(data whitening in } \mathbf{\pi}_d)$$

(7)

$$\mathbf{W}_d = \mathbf{W}_d (\text{rescaling by } q_d)$$

(8)

$$\mathbf{W}_d = \mathbf{U}_d \mathbf{W}_d + \mathbf{R} \quad (\text{adding residuals})$$

(9)

Finally, considering that the updated estimates the whitened means $\bar{\mathbf{w}}_k$, obtained by “partially whitening” the class means $\bar{\mathbf{w}}_k$ and $\bar{\mathbf{w}}_a$, by the inner product $\mathbf{w} \cdot \mathbf{p}$, where $\mathbf{w}$ is computed as follows:

$$\mathbf{w} = \mathbf{W}_d \mathbf{f} = q_d \mathbf{U}_d \mathbf{Q}_d^{-1} \mathbf{U}_d^T \mathbf{f} + \mathbf{U}_d^T \mathbf{R} \cdot \mathbf{f}$$

(12)

Finally, given $\gamma$ as in Eqs. (5) and (6), $\mathbf{p}$ is assigned to class $A$ if $\mathbf{w} \cdot \mathbf{p} < \gamma$; to class $B$ otherwise.

Notice that we never explicitly compute the matrix $\mathbf{W}$, but we perform the matrix times vector operations reported in Eqs. (1) and (12), thus preventing a quadratic time and space complexity. After the training phase, the classification model is represented by $\mathbf{w}$ and $\gamma$. In the following, the described batch classifier is called truncated $\text{IPCAC} (=\text{O-IPCAC})$ to differentiate it from its extended versions, called $\text{O-IPCAC}$, designed to perform online/incremental training.

O-IPCAC is necessary when the cardinality of the training set is too high, or when mini-batches of training data $\mathbf{B}_k = \mathbf{B}_{A,k} \cup \mathbf{B}_{B,k}$ are dynamically supplied, so that subsequent training phases must be applied to update the classification model. To this aim, O-IPCAC works by continuously updating the following parameters:

$$N_k, N_{A,k}, N_{B,k}$$

number of training points seen until the $k$-th training phase;

$$\bar{\mathbf{w}}_k, \bar{\mathbf{w}}_{A,k}, \bar{\mathbf{w}}_{B,k}$$

the means in the original space, which are employed to obtain the centered sets $\bar{\mathbf{P}}_k$, $\bar{\mathbf{P}}_{A,k}$, and $\bar{\mathbf{P}}_{B,k}$, respectively;

$\mathbf{U}_d, \mathbf{Q}_d, \mathbf{V}_d$ the $\text{SVD}$ matrices related to $\mathbf{P}_k$, truncated to $d$ principal components;

the standard deviations of the projections $\mathbf{w}_d^T \mathbf{P}_A$ and $\mathbf{w}_d^T \mathbf{P}_B$.

As in Eqs. (5) and (6), we perform the matrix times vector operations reported in Eqs. (1) and (12), thus preventing a quadratic time and space complexity. After the training phase, the classification model is represented by $\mathbf{w}$ and $\gamma$. In the following, the described batch classifier is called truncated $\text{IPCAC} (=\text{O-IPCAC})$ to differentiate it from its extended versions, called $\text{O-IPCAC}$, designed to perform online/incremental training.

Consider the case when $k-1$ training phases have been performed and a new mini-batch $\mathbf{B}_k = \mathbf{B}_{A,k} \cup \mathbf{B}_{B,k}$ is provided (where $\mathbf{B}_{A,k}$ and $\mathbf{B}_{B,k}$ are the two subsets containing points belonging to classes $A$ and $B$, respectively), where $|\mathbf{B}_k| = n_k = n_{A,k} + n_{B,k}$. Firstly, we update the training set cardinalities:

$$N_k = n_{k-1} + n_k, N_{A,k} = N_{A,k-1} + n_{A,k}, N_{B,k} = N_{B,k-1} + n_{B,k}$$

the mean vectors:

$$\mu_k = \frac{\mu_{k-1} N_{k-1} + \sum_{p \in \mathbf{B}_k} \mathbf{p}}{N_k}$$

(11)

and we center the points in $\mathbf{B}_k$ around the old mean: $\mathbf{\bar{B}}_k = \mathbf{\bar{B}}_{k-1} \cup \mathbf{\bar{B}}_k$.

Secondly, we update the $\text{TSVD}$ matrices\footnote{We underline that the number of retained components $d_k = \min(\log_2 N_k, D)$ might increase at each step, and the rank of the computed $\text{TSVD}$ matrices would grow accordingly, up to the maximum value $D$.} by means of the algorithm described in [58] and by exploiting the information carried by $\mathbf{\bar{B}}_k$, that is: $\mathbf{\bar{P}}_k = U_d \mathbf{Q}_d \mathbf{V}_d^T$, where $\mathbf{\bar{P}}_k$ represents the set $\mathbf{\bar{P}}_k = \mathbf{\bar{P}}_{k-1} \cup \mathbf{\bar{B}}_k$.

Finally, considering that the updated $\text{TSVD}$ matrices are related to the points $\mathbf{\bar{P}}_k$ that are centered on $\mu_{k-1}$, a re-centering operation is required to obtain $\text{TSVD}$ matrices related to points centered on $\mu_k$; this is done by applying the re-centering rank-one modification described in [58], choosing as translation factor the quantity $(n_k/N_k) \times \langle \mathbf{\bar{E}}_k \rangle$, where $\langle \mathbf{\bar{E}}_k \rangle$ is the centroid of points in $\mathbf{\bar{B}}_k$.

Given the updated means and $\text{TSVD}$ matrices, we can:

- estimate the whitened means $\bar{\mathbf{w}}_{A,k}$ and $\bar{\mathbf{w}}_{B,k}$ by employing Eq. (11);
- obtain the updated vector $\mathbf{f}_k$;
- compute the new vector $\mathbf{w}_k$ by means of Eq. (12).

Please cite this article as: A. Rozza, et al., Novel Fisher discriminant classifiers, Pattern Recognition (2012), http://dx.doi.org/http://dx.doi.org/10.1016/j.patcog.2012.03.021
Notice that these computations are required to store only \( O(Dn_0 + Dd_k) \) real values per mini-batch.

Regarding the update of the thresholding value \( \gamma \), we have not employed Eq. (5), since it requires to store the whole training set, and it is not able to handle unbalanced classes. Therefore, we adopted the thresholding value reported in Eq. (6), since it is more effective and it can be easily updated by storing only few information.

Considering that \( \sigma^2 = E[x^2] - E[x]^2 \), the quantities employed in Eq. (6) are updated as follows:

\[
\mathcal{P}_{Ak} = \frac{N_{Ak-1}\mathcal{P}_{Ak-1} + n_{Ak}\langle w_i^TB_{Ak}\rangle}{N_{Ak}}
\]

\[
\mathcal{P}_{2Ak} = E[(w_i^TB_{2Ak})^2 | p \in \mathcal{P}_{Ak-1}] = \mathcal{P}_{2Ak-1} + \sigma^2_{Ak-1}
\]

\[
\mathcal{P}_{2Ak} = E[(w_i^TB_{2Ak})^2 | p \in \mathcal{P}_{Ak}] = \frac{N_{Ak-1}\mathcal{P}_{2Ak-1} + n_{Ak}\langle (w_i^TB_{Ak})^2 \rangle}{N_{Ak}}
\]

\[
\sigma_{Ak} = \sqrt{\mathcal{P}_{2Ak} - \mathcal{P}_{Ak}^2}
\]

where we have defined the set of real values:

\[
(w_i^TB_{Ak})^2 = \langle w_i^TB_{Ak}\rangle^2 | p \in \mathcal{P}_{Ak}
\]

The updated quantities \( \mathcal{P}_{Ak} \), \( \mathcal{P}_{2Ak} \), and \( \sigma_{Ak} \) are similarly evaluated.

After the \( k \)-th update of the trained model, the described online algorithm approximates the Fisher subspace with respect to the points in \( \mathcal{P}_k \), and these points will no longer be needed for future updates; therefore, O-IPCAC performs only one pass through the data to compute its classification model. We would finally like to note that the classification model computed by O-IPCAC after the analysis of \( k \) mini-batches of training data \((B_1, B_2, \ldots, B_k))\), is equal to the one that would be computed by applying T-IPCAC on the whole training set \( \mathcal{P}_k \); therefore, when the data dimensionality and/or cardinality allows to analyze the whole training set at once, the computed T-IPCAC model equals the one that would be obtained by O-IPCAC, and the same classification performance is achieved.

The computational cost of the training phase is dominated by the incremental rank-d singular value decomposition that requires \( O(Dn_0) \) operations for \( d \leq \sqrt{\min(n, D)} \) [58]. Regarding the memory requirements, O-IPCAC stores \( O(Dn_0 + Dd_k) \) real values during the training tasks, and only \( O(D) \) values during classification.

### 6. Kernel versions of IPCAC

The classification algorithms proposed in the previous sections cannot deal with non-linearly separable classes; to overcome this weakness we have developed their kernel versions, K-IPCAC and K-\( \text{KPCA} \), which are described in this section.

To relax the linear separability constraint imposed by the IPCAC algorithm, we exploited the kernel trick as in the kernel principal component analysis (KPCA), thus obtaining a kernel isotropic principal component analysis classifier (K-\( \text{KPCA} \)).

The main idea is that the classes to be separated are non-linearly separable in the original space \( \mathcal{Q}^D \subseteq \mathbb{R}^D \), but they are linearly separable in a higher dimensional space \( \mathcal{Q}^h \) that is reachable through a non-invertible map \( \phi(x) \). The training points in the original space, that is \( p_i \in \mathcal{P}_k^m \), \( \mathcal{P}^m \subseteq \mathbb{R}^D \), could then be separated by first applying the PCA on their projections \( \phi(p_i) \in \mathcal{Q}^h \) after which a Fisher subspace will be estimated. More precisely, the PCA step would compute a set of \( N \) relevant principal components \( \{x_k^h\}_{k=1}^N \). The subspace spanned by the vectors \( \{x_k^h\}_{k=1}^N \) is the KPCA subspace where the vectors \( \phi(p_i) \) are finally projected. To this aim, in [37] the authors prove that, given a kernel function \( k(p, p_i) \) that allows to compute the dot product between \( \phi(p) \) and \( \phi(p_i) \), it is possible to compute a weight matrix \( A = [x_k^h]_{k=1}^N \) that allows to calculate the projection of a point \( \phi(p) \) on the principal components \( \{x_k^h\}_{k=1}^N \). Assuming that the mapped points \( \phi(p_i) \) are mean centered in \( \mathcal{Q}^h \), A is computed as follows:

1. compute the Gram matrix \( K = [k(p_i, p)]_{i=1}^N \) where \( \{p_i\}_{i=1}^N \) are the training vectors;
2. compute the eigendecomposition on the design matrix \( \mathcal{K} = A^TA \), and remove eventual zero-variance components obtaining the new decomposition \( \mathcal{K} = \mathcal{A}\mathcal{A}^T \), where \( \mathcal{N} \) components are retained;
3. compute the weight matrix \( A = \mathcal{A}^{-1/2} \).

Having computed \( A \), the generic point \( \phi(p) \in \mathcal{Q}^h \) can be projected on \( \{x_k^h\}_{k=1}^N \) as:

\[
\mathcal{K} = K - I_N \mathcal{K} B_1 + B_1 \mathcal{K} - B_1 B_1^T
\]

where \( I_N = [N_{11}]_{i=1}^N \), that is a \( N \times N \) matrix with all entries equal to \( 1/N \).

Exploiting these theoretical results, we derived a method to compute the Fisher subspace on training data projected on the KPCA subspace. To describe our method we consider a training set mean centered in \( \mathcal{Q}^h \); as can be noted, Eq. (13) can be restated in matrix form as:

\[
K = \mathcal{K} - I_N \mathcal{K} B_1 + B_1 \mathcal{K} - B_1 B_1^T
\]

Next, we represent the projection of the training points \( \mathcal{P} \subset \mathcal{Q}^D \) on the principal components \( \{x_k^h\}_{k=1}^N \subset \mathcal{Q}^h \) with a matrix \( \mathcal{P} \) obtained as follows:

\[
\mathcal{P}_k = [\langle x_k^h \phi(p) \rangle_{k=1}^N] = A^{-1/2}A^T \mathcal{K}
\]

\[
A = \mathcal{A}^{-1/2} \mathcal{A}^{-1/2}^T \mathcal{A} \mathcal{A}^T = N^{-1/2} \mathcal{A}^T
\]

being \( \mathcal{A} = \mathcal{I} \) for the orthogonal of \( \mathcal{A} \).

Finally, the Fisher subspace is calculated by employing the class means of the points represented by the columns of \( \mathcal{P} \).

To relax the hypothesis about mean centering of the training points in \( \mathcal{Q}^h \), we consider the projections of the centered points \( \mathcal{P}(\phi(p) - \mu_\phi)_{i=1}^N \) and we exploit Eq. (14). More precisely, calling \( \mu_\phi = N^{-1} \sum_i \phi(p_i) \) the mean of the training points mapped in \( \mathcal{Q}^h \).
we compute the matrix $P_{\phi}$ as follows:

$$P_{\phi} = \begin{bmatrix} \Lambda_k^{(1/2)}N_k \cdot (\phi(p_k) - \mu_{\phi})^{N_k} \\ \vdots \\ \Lambda_k^{(1/2)}N_k \cdot (\phi(p_k) - \mu_{\phi})^{N_k} \end{bmatrix}_{i=k=1}^{N_{\mathcal{F}}}$$

$$= \begin{bmatrix} \sum_{i=1}^{N_k} \Lambda_k^{(1/2)}N_k \cdot (\phi(p_{ki}) - \mu_{\phi})^{N_k} \\ \vdots \\ \sum_{i=1}^{N_k} \Lambda_k^{(1/2)}N_k \cdot (\phi(p_{ki}) - \mu_{\phi})^{N_k} \end{bmatrix}_{i=k=1}^{N_{\mathcal{F}}}$$

Note that the results reported in Eqs. (16) and (17) differ only in the definition of $\tilde{A}$ and $\tilde{A}$; indeed, in Eq. (16) $\tilde{A}$ and $\tilde{A}$ are the decomposition matrices of $\mathbf{K}$, while in Eq. (17) they are the decomposition matrices of the centered design matrix $\tilde{K}$.

Assuming that the first $N_\mathcal{A}$ column vectors $P_{\phi} \mid \mathcal{A}, P_{\phi} \mid \mathcal{B}$ belong to class $A$, and that the remaining column vectors $P_{\phi} \mid N_{\mathcal{A}} = \mathcal{B}$ belong to class $B$, and noting that these points are whitened through the KPCA algorithm, it is possible to use them for a direct Fisher subspace estimation. To this aim, we must compute the quantities:

$$\mu_{\mathcal{A} \phi} = \langle \phi(p_{\mathcal{A}}) - \mu_{\phi} \rangle_{i=1}^N, \quad \mu_{\mathcal{B} \phi} = \langle \phi(p_{\mathcal{B}}) - \mu_{\phi} \rangle_{i=1}^N$$

and their difference:

$$\tilde{J} = \langle \phi(p_{\mathcal{A}}) - \mu_{\phi} \rangle_{i=1}^N \cdot \langle \phi(p_{\mathcal{B}}) - \mu_{\phi} \rangle_{i=1}^N$$

$$= P_{\phi} \mid \mathcal{A}, P_{\phi} \mid \mathcal{B} \cdot \begin{bmatrix} N_{\mathcal{A}} \cdots N_{\mathcal{A}} \\ \vdots \\ N_{\mathcal{B}} \cdots N_{\mathcal{B}} \end{bmatrix}^T - P_{\phi} \mid \mathcal{A}, P_{\phi} \mid \mathcal{B} \cdot \begin{bmatrix} 0 \cdots 0 \\ \vdots \\ 0 \cdots 0 \end{bmatrix}$$

$$= P_{\phi} \mid \mathcal{A}, P_{\phi} \mid \mathcal{B} \cdot \begin{bmatrix} N_{\mathcal{A}} \cdots N_{\mathcal{A}} \\ \vdots \\ N_{\mathcal{B}} \cdots N_{\mathcal{B}} \end{bmatrix}^T$$

where we have defined:

$$N_{\mathcal{A} \mid \mathcal{B}} = \begin{bmatrix} N_{\mathcal{A}} \cdots N_{\mathcal{A}} \\ \vdots \\ N_{\mathcal{B}} \cdots N_{\mathcal{B}} \end{bmatrix}^T$$

The vector $\tilde{J}$ must be normalized; its norm is:

$$||\tilde{J}|| = \sqrt{N_{\mathcal{A} \mid \mathcal{B}}} \cdot \left( ||\mathbf{N}_{\mathcal{A} \mid \mathcal{B}}||_2 \right)^2$$

thus, the Fisher subspace can be computed as follows:

$$\mathbf{f} = \tilde{J}/||\tilde{J}|| = -N_{\mathcal{A} \mid \mathcal{B}} \cdot \Lambda^{1/2} \cdot N_{\mathcal{A} \mid \mathcal{B}}^T$$

In particular, if $N_{\mathcal{A}} = N_{\mathcal{B}}$ we get:

$$\mathbf{f} = \frac{\tilde{J}}{||\tilde{J}||} = -\frac{1}{N_{\mathcal{A}}} \cdot \begin{bmatrix} 1 \cdots 1 \\ -1 \cdots -1 \end{bmatrix}^T$$

Given a test point $\mathbf{p}$, we compute its projection on $\mathbf{f}$ by means of Eq. (15) and Eq. (18):

$$\text{proj}(\mathbf{p}) = (\Lambda^{1/2} \cdot \mathbf{Ker})^T \cdot (\mathbf{f})$$

$$= (\mathbf{Ker})^T \cdot \Lambda^{1/2} \cdot \Lambda^{1/2} \cdot N_{\mathcal{A} \mid \mathcal{B}}^T$$

$$= \mathbf{Ker}^T \cdot N_{\mathcal{A} \mid \mathcal{B}}$$

Note that, since the weight vector $\mathbf{w}$ can be precomputed at training time, the classification algorithm is similar to that obtained by Eq. (4), that is $\mathbf{w} \cdot \mathbf{Ker} > \gamma$, where the thresholding value $\gamma$ is estimated by employing either Eqs. (5) or (6).

During the classification of a test point, the obtained classifier requires only $N$ kernel function evaluations more than the IPCAC algorithm, thus remaining very simple and efficient. In Section 3 we recalled that KFD is equivalent to KPCA plus LDA [39]; K-IPCAC offers a similar representation (KPCA+IPCAC), guaranteeing an approach that reduces the time and space requirements with respect to the sequential application of the two techniques.

6.1. Kernel truncated IPCAC

In Section 6 we show that a point $\mathbf{p}$ can be projected on the Fisher subspace computed in the kernel space by employing Eq. (19).

In this section we present an improvement of K-IPCAC, called K-TIPCAC, that has been developed to reduce the overfitting problems that could affect kernel methods.

To this aim, K-TIPCAC exploits the same concept at the basis of the algorithm presented in Section 5, and whitens the data in the linear subspace $\pi_{B} = \text{Span} \{ \mathbf{v}_1, \ldots, \mathbf{v}_d \}$, spanned by the first $d$ principal components in $\mathbf{Q}_B$, while maintaining unaltered the information related to the orthogonal subspace.

More precisely, after the diagonalization of the Gram-matrix described in step 2 of Section 6, we select the $d$ largest eigenvalues, that represent a fixed amount of variance defined a-priori, and we set the remaining part of the spectrum to 1.

This process reduces the overfitting problems produced by the smallest, and more noisy, part of the spectrum without losing any discriminative information. The advantages of employing K-TIPCAC are supported by our experimental results reported in Section 7.

Notice that, both K-IPCAC and K-TIPCAC are $O(N^3)$ in training and $O(ND)$ in classification time complexity, while they require to store $O(ND)$ values.

7. Algorithm evaluation

Due to the promising results achieved on two binary classification problems concerning spam email recognition and EEG signal classification (see [18,19] or the supplementary materials for detailed descriptions), which are characterized by high dimensional data, small sample size, and unbalanced classes, in this section we present the experiments performed to evaluate the quality of the proposed methods when facing multiclass classification problems.

Note that, since T-IPCAC obtains the same results as O-IPCAC, the only difference being the time and space complexity, in the following experiments we only show the results achieved by T-IPCAC (anyhow, experiments employing O-IPCAC can be found in the supplementary material).

Before presenting our experiments, we describe two techniques we employed to combine several binary classifiers to obtain ensemble methods for multiclass classification (see Section 7.1); secondly, we present the results we obtained by applying the developed techniques on two multiclass image classification problems, which are face recognition (see Section 7.2), and hand written digit recognition (see Section 7.3). We note that, as mentioned in Section 2, the two benchmark databases employed for face recognition are characterized by a number of training examples that is much lower than the space dimensionality. Regarding the hand written digit recognition problem, though the employed training database has a high cardinality, we have run our tests by selecting a small subset to have a training set cardinality comparable to the data dimensionality.

---

7 The value of $d$ is chosen so that $(\sum_{i=1}^d \xi_i)/\sum_{i=1}^N \xi_i$ equals the desired proportion of variance var to be captured; in our experiments the value of var is empirically set.
7.1. Ensemble methods for multiclass classification

7.1.1. Decision DAGs (DDAGs)

A rooted direct acyclic graph (DAG) is a graph whose edges have an orientation, no cycles, and only one root node. A rooted binary DAG has nodes which have either 0 or 2 arcs leaving them. A DDAG [59] is a method that combines the results of one-against-one classifiers to produce a multiclass classification. To this aim, considering a C-class classification problem, the DDAG is implemented using a rooted binary DAG with \( K = C(C-1)/2 \) internal nodes. Each node represents a classification model trained on two of the C classes, and it produces a Boolean output value, that is, a value \( v \in \{0, 1\} \). The nodes are arranged in a binary tree with the single root node at the top, two nodes in the second layer and so on until the final layer of leaves, which represent all the classes to be discriminated. Considering each classifier as a Boolean function, to classify an unknown sample the DDAG proceeds as follows: it starts at the root node and it evaluates the Boolean function; the node is then exited either via the left edge, if \( v \) equals 0, or via the right edge, if \( v = 1 \). The next node's binary function is then evaluated, and the procedure iterates until a leaf node is reached. The sample is then assigned to the class associated to the reached leaf node.

Given an input dataset composed of points belonging to \( C \) classes, and having selected one of our binary classifying models (that is, T-IPCAC or K-TIPCAC) as the engine classifier, we train \( K \) binary classifiers, each discriminating two classes in a one-against-one fashion, and we combine their results by means of DDAG; the ensemble multiclass classification machines thus obtained will be referred as DDAG T-IPCAC and DDAG K-TIPCAC in the following.

7.1.2. Pairwise coupling with majority vote (PWCMV)

Given an input dataset composed of points belonging to \( C \) classes, and having selected one of our binary classifying models (that is, T-IPCAC or K-TIPCAC) as the engine classifier, pairwise coupling with majority vote (PWCMV) requires to train \( K = C(C-1)/2 \) binary classifiers, each discriminating two classes in a one-against-one fashion. Considering the decision of each binary classifier as a “vote” for a specific class, an unknown sample is firstly classified with all the trained \( K \) binary classifiers, and it is then assigned to the class that has achieved the highest number of votes. In the following, the ensemble machines obtained by employing either T-IPCAC or K-TIPCAC as the engine classifiers will be referred as PWCMV T-IPCAC and PWCMV K-TIPCAC.

7.2. Experiments on face recognition

Within the last decades, 2D face recognition algorithms have been widely investigated due to both commercial and security needs. A detailed survey of 2D face recognition algorithms can be found in [60]; in this survey the authors stress that the problem of face recognition is made particularly difficult by the fact that the number of samples (images) per class (person) is very limited compared to the feature space dimensionality (small sample size problem). For this reason, a growing interest has been shown in dimensionality reduction methods specifically designed for this task, which assume that face images, represented as high-dimensional pixel arrays, often belong to a subspace of intrinsically low dimension. To estimate this subspace, and to discriminate sample points after their projection on it, different methods have been proposed, among which Eigenfaces [61] and Fisherfaces [30], based on linear PCA and LDA, respectively, are the most cited examples. Moreover, other interesting face recognition methods are those presented in [45,46], where LDA is applied after representing the images by means of the Ext-Isomap method, whose strength is the substitution of the Euclidean distance between sample points with the geodesic one. Though promising results have been obtained by the cited techniques, their recognition rate has been much improved [62] by introducing kernel methods into PCA and LDA, thus obtaining the kernel-Eigenfaces method [40], which essentially exploits KPCA [37], the kernel-Fisherfaces method [43], which essentially exploits KFD [42], and the KFD-Isomap method [44], which substitutes LDA with KFD.

To critically assess our techniques in the following we compare the face recognition rates we computed with those achieved by KFD [42] and KFD-Isomap [44]; we chose these kernel methods as baseline algorithms since they are strongly related to our work, and have performed reproducible experiments on two benchmark datasets. Furthermore, we note that in their work the authors report the results achieved, under the same experimental settings, by Isomap [47], Ext-Isomap [45,46], Eigenfaces [61], Fisherfaces [30], and KPCA [37]. This allows us to perform a further comparison with these methods.

To accomplish face recognition, we employed both T-IPCAC and K-TIPCAC (with either the polynomial kernel or the RBF kernel) as engine algorithms of DDAG and PWCMV, thus obtaining four ensemble classifiers, that are DDAG T-IPCAC, DDAG K-TIPCAC, PWCMV T-IPCAC, PWCMV K-TIPCAC; all the T-IPCACs and K-TIPCACs employ the threshold value computed as described in Eq. (6).

Regarding the parameter values, we recall that T-IPCAC requires only the dimensionality \( d \) of the space \( \mathbb{R}^d \) where the partial whitening is performed (see Section 5), while K-TIPCAC automatically chooses the value of the parameter \( d \) according to the value of the parameter \( \var{\text{var}} \) (see Section 6.1), which is the data variance to be retained, and it employs either a polynomial kernel of degree deg, or a RBF kernel with a parameterized variance \( \sigma \).

To obtain an unbiased evaluation scheme, all the classification experiments have been repeated \( n \) times and the values of the parameters have been varied to achieve the lowest error rates in each run. This parameter setting procedure is that employed in [42,44], where the values of \( n \) are varied in our experiments to be as similar as possible to those reported by the authors. We note that this approach could lead to bias because the more parameters a method has, the more it can overfit. However, since the number of parameters between methods seems fairly similar this might not be a large concern.

The two benchmark face databases employed in the experiments are the AT&T\(^8\) and the Yale\(^9\) face databases, which are described in Sections 7.2.1 and 7.2.2.

7.2.1. Experimental settings and results on AT&T database

The AT&T face database contains 400 gray scale images, each of size 92 × 112 pixels, of 40 subjects (10 images per subject), including 36 males and 4 females. It includes variations in pose and scale, facial expression (open/closed eyes, smiling/not smiling), facial details (glasses/no glasses, beard and mustaches), and slight illumination variations. In Fig. 2—top we show the images of one female subject and two male subjects.

To compare our results to those of KFD [42], we performed experiments representing each image as a vector of 10,304 elements containing the pixel gray levels, and avoiding any image preprocessing step.

\(^8\) The AT&T database is publicly available at http://www.uk.research.att.com/facedatabase.html.  
\(^9\) The Yale database is publicly available at http://cvc.yale.edu/projects/yalefaces/yalefaces.html.
The classification experiments were repeated 20 times, and each time five images per person were randomly selected for training, while the remaining five were employed for testing. For this experiment our kernel ensembles employed the polynomial kernel and we varied the value of its degree as it has been described in [42] (tests were performed with the values $\deg = 2, 3, 4$).

The error rates computed by the four ensemble methods we experimented, averaged over the 20 iterations, are reported in the top of Table 1; for a direct comparison, in this table we also show the performance, reported in [42], of KFD and KPCA, while in the second column we report the values of the parameters that, on the average of the executed iterations, allowed to achieve the best results (the parameter value of KFD and KPCA is the degree $\deg$ of the polynomial kernel they employ).

Considering the results reported in the top of Table 1, the reader can note that the four kernel methods achieve the best performance. Among them, our kernel ensemble machines outperformed both KFD and KPCA. To further show the good quality of our classifiers, at the bottom of Table 1 we report the results achieved by the experiments that were aimed at comparing the performance of our kernel classifiers to that of KFD and KPCA, when the degree of the polynomial kernel is fixed to the three values $\deg = 2, 3, 4$ (we underline that the results of KFD and KPCA are those reported in [42]). These results show that, when the degree of the polynomial kernel is fixed, our ensembles always outperform the other two methods.

The second experiment we performed is aimed at comparing the results achieved by our algorithms to those of KFD-Isomap [44]; therefore, we performed experiments by firstly applying the same preprocessing steps. More precisely, we resized each face image to $46 \times 56$ pixels, and we represented it by a raster scan vector of intensity values, normalized to be a zero-mean and unit-

\[ \begin{align*}
\text{AT&T Face Database} \\
\text{Yale Face Database} \\
\text{MNIST Database}
\end{align*} \]

Fig. 2. Example images from the AT&T face database, the Yale face database, and the MNIST hand written digit database (note that, for better visualization, we showed the negative MNIST images).
variance vector. The samples thus computed were randomly divided into two disjoint sets: six images per person for training and the remaining four for test. Moreover, since in [44] the authors reduce the training time complexity by projecting the points in a reduced r-dimensional space, to perform a fair comparison we also introduced a dimensionality reduction step in our T-IPCAC and K-TIPCAC binary classifiers. This step is described in the following for the general case of C classes to be discriminated.

To perform dimensionality reduction, we firstly evaluate the Fisher subspace of the overall C classes by generalizing their “partial whitening” approach. To this aim, after the partial whitening, the whitened class means \( \mu_{c, i}^{w} \) are computed as follows:

\[
\mu_{c, i}^{w} = W \mu_{c, i} = (Q_{i} \mu_{c, i} Q_{i}^{T} + I - U_{c, i} U_{c, i}^{T}) \mu_{c, i}
\]

where \( \mu_{c, i} \) is the class mean in the original space, and the employed matrices and quantities are explained in details in Section 5 and Eq. (11). At this stage the orthonormal basis, \( \Pi_{c, i} \), composed of C−1 vectors spanning the Fisher subspace, can be computed. More precisely, \( \Pi_{c, i} \) is obtained by orthonormalizing the \( C - 1 \) linearly independent whitened mean vectors \( \mu_{c, i} \) through the Gram–Schmidt procedure. The partially whitened training vectors, stored in the matrix \( P_{\Pi_{c, i}} \), computed as described in Eq. (7), are then projected on the subspace \( \Pi_{c, i} \), obtaining the set of \( (C - 1) \)-dimensional points

\[
\mathcal{P}_{\Pi_{c, i}} = \{ F S_{i} \mathbf{p} | \mathbf{p} \in \mathcal{P}_{\Pi_{c, i}} \}
\]

where \( \mathbf{F} \) is the matrix whose columns span the Fisher subspace.

Exploiting the points in \( \mathcal{P}_{\Pi_{c, i}} \), \( (C - 1)/2 \) binary (T-IPCAC or K-TIPCAC) classifiers can be trained and combined by exploiting either DDAG or PWCMV. We recall that the AT&T dataset comprises C≈40 classes, and all the (training and test) points are therefore projected on the 39-dimensional Fisher subspace estimated from the training data.  

To obtain an unbiased evaluation, we performed this experiment 400 times and we averaged the obtained results. Under this experimental setting we tested both our linear ensembles (DDAG T-IPCAC and DDAG K-TIPCAC), and the kernel ones (PWCMV T-IPCAC and PWCMV K-TIPCAC), where all the K-TIPCACs employ the RBF kernel of variance \( \sigma \), whose value is experimentally chosen. The average error rates achieved by the four ensemble machines are reported in the top of Table 2; for a direct comparison, in this table we also show the performance, as they were reported in [44], of Eigenface, Fisherface, Isomap, Ext-Isomap, KFD-Isomap, while in the second column we report, for each algorithm, the parameter values which, on average, allowed to achieve the best average performance over the 400 runs (every algorithm cited in [44] needs the dimension \( r \) of the reduced space, the algorithms employing Isomap also need the radius \( \epsilon \) of the neighborhood where they search for the closest point during the geodesic distance estimation, and KFD-Isomap needs the degree \( deg \) of the employed polynomial kernel). 

As in the previous experiments, we note that our kernel techniques achieve the lowest error rate.

### 7.2.2. Experimental settings and results on Yale database

The Yale face database contains 165 gray scale images, each of size 320×243 pixels, of 15 subjects (11 images per subject), among which 14 are males and 1 is female, in a variety of facial expressions and pose conditions, including images with and without glasses, and strong illumination variations.

To compare the results achieved by our algorithms to those of KFD-Isomap [44], the images were similarly preprocessed. More precisely, since the images in the database have large background, they were cropped by extracting a sub-image with size 192×236 pixels that includes only the facial contours. Furthermore, each extracted sub-image was resized to 48×59. In Fig. 2—center we show face images from the Yale database: in the top a face image before cropping, and after cropping is shown, while in the bottom we show closely cropped images of a male subject and a female one.

As we explained in the second experiment on AT&T database, which has the same experimental setting, each (closely cropped and resized) image in the Yale dataset was represented by a raster scan image.
Automatic recognition of handwritten characters has been one of the most investigated problems in the field of pattern recognition. Although this is nowadays a solved problem, the existence of several benchmark databases comprising several images of handwritten characters explains the great deal of pattern recognition algorithms. Although this is nowadays a solved problem, the existence of several benchmark databases comprising several images of handwritten characters explains the great deal of pattern recognition algorithms. Furthermore, we note that the performance of all the compared algorithms is worse than that achieved, under the lowest error rate. Moreover, we reproduce the same experimental setting, by representing each image by a raster scan vector of the same database; therefore, we reproduce the same experimental settings, by representing each image by a raster scan vector of the same database; this is because the images in the Yale database are much more varied, and are therefore more difficult to be learnt. Nevertheless, while the performance drop-off of Eigenface, Fisherface, Isomap, Ext-Isomap, and KFD-Isomap is quite high, our techniques are less affected. This proves the quality of our ensembles.

7.3. Experiments on handwritten digit recognition

Automatic recognition of handwritten characters has been one of the most investigated problems in the field of pattern recognition. Although this is nowadays a solved problem, the existence of several benchmark databases comprising several images of handwritten characters explains the great deal of pattern recognition algorithms whose performance is assessed by testing them on handwritten characters recognition. One of the most used benchmark datasets containing handwritten digits is the MNIST database, which contains a training set of 60,000 example images, and a test set of 10,000 example images. Considering that each handwritten digit can be viewed as a connected region in the image, the database has been created by normalizing the sizes of all the connected regions, while preserving their aspect ratio (e.g., elongation, compactness), and each region has been centered in a 28 × 28 gray scale image; this last step has been performed by translating the region so that the center of mass of its pixels is at the center of the 28 × 28 gray scale image. Some (negatives of) handwritten digit images of the MNIST database are shown in Fig. 2—bottom.

To better assess the efficacy of our classifiers we compared their results to those achieved by Ext-Isomap [45] when employing the same database; therefore, we reproduced the same experimental settings, by representing each image by a raster scan vector of intensity values without applying any preprocessing or feature extraction algorithm. Moreover, we run tests by employing a randomly selected training set of 1500 MNIST images and a non-overlapping test set of 250 images.

To accomplish this multiclass classification task, we tested DDAG T-IPCAC, PWCMV T-IPCAC, DDAG K-IPCAC, and PWCMV K-IPCAC, where the kernel engine classifiers employed the RBF kernel, and all the engine algorithms employ the threshold value computed as described in Eq. (6). Regarding the parameter values, K-IPCAC requires the value of the variance \( \sigma^2 \) to be retained, and the value of the standard deviation \( \sigma \) of the RBF kernel to be computed as described in Eq. (6).

As in the previous experiments, our kernel ensembles achieve the lowest error rate. Moreover, we note that the performance of all the compared algorithms is worse than that achieved, under the lowest error rate. Moreover, we reproduce the same experimental settings, by representing each image by a raster scan vector of the same database; this is because the images in the Yale database are much more varied, and are therefore more difficult to be learnt. Nevertheless, while the performance drop-off of Eigenface, Fisherface, Isomap, Ext-Isomap, and KFD-Isomap is quite high, our techniques are less affected. This proves the quality of our ensembles.

8. Conclusions and future works

In this work we propose novel classifiers based on a new approach to estimate the Fisher subspace. We faced different classification tasks to stress our methods, showing their qualities in the following situations: reduction of the effects caused by the small sample size problem, handling of high dimensional data, management of classification tasks where the space dimensionality is bigger than, or comparable to, the cardinality of the training set, and the classes are non-linearly separable.

The proposed classifiers, some of which are also described in [18,19,17], obtain results that are usually better than those achieved by well-known classifiers. These results are of particular importance because they demonstrate that our methods are valuable tools for real data classification, and they offer the possibility to be employed in a wide range of applications.

In future works, we want to improve both T-IPCAC (see Section 5) and K-IPCAC (see Section 6.1) by developing a time efficient technique to automatically evaluate the dimensionality of the subspace \( p_s \) where the “partial whitening” is performed; moreover, we plan to bound the error rate of the online learning algorithm (O-IPCAC). Finally, we will focus on the development of the online version of the K-IPCAC technique, to deal with non-linearly separable data dynamically supplied.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2012.03.021.

References

Alessandro Rozza obtained his master degree cum Laude, from the Department of Computer Science of Università degli Studi di Milano-Bicocca in the 2006. He received the Ph.D. degree in Computer Science in March 2011 from the Department of Science dell’Informazione, Università degli Studi di Milano. His research interests include machine learning and bioinformatics.

Gabriele Lombardi obtained his master degree cum Laude from the Department of Computer Science of Università degli Studi di Milano in the 2004. In the same University he obtained his Ph.D. in Applied Mathematics in the 2009. His research interests include machine learning, manifold learning, and computer vision.

Elena Casiraghi graduated cum Laude in Biological Sciences from Università degli Studi di Modena in the 1975. In 1988 she obtained her Ph.D. in Computer Science from the Università degli Studi di Milano. Currently she is a dean of Faculty of Mathematical, Fisical and Natural Science, Università degli Studi di Milano. Her scientific interests range from machine learning, pattern recognition, and image processing.

Paola Campadelli graduated cum Laude in Biological Sciences from Università degli Studi di Modena in the 1975. In 1988 she obtained her Ph.D. in Computer Science from the Università degli Studi di Milano. Currently she is a dean of Faculty of Mathematical, Fisical and Natural Science, Università degli Studi di Milano. Her scientific interests range from machine learning, pattern recognition, and image processing.