Principal Component Analysis for Supervised Learning: a minimum classification error approach

Tiago Buarque Assunção de Carvalho[1], Maria Aparecida Amorim Sibaldo[1], Ing Ren Tsang[2], George Darmiton da Cunha Cavalcanti[2]

[1] Universidade Federal Rural de Pernambuco, Brasil
{tiago.buarque, mariaaparecida.sibaldo}@ufrpe.br
[2] Universidade Federal de Pernambuco, Brasil
{tir,gdcc}@cin.ufpe.br

Abstract.

We present an alternative method to use Principal Component Analysis (PCA) for supervised learning. The proposed method extract features similarly to PCA, but the features are selected by minimizing the Bayes error rate for classification. We show that the proposed method selects features that best separate the elements of the different classes. Using real datasets, along with four different classifiers, experimental results show that the recognition accuracy of the proposed technique is improved compared to PCA.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning.

Keywords: Principal component analysis, Dimensionality reduction and manifold learning, Supervised learning by classification, Data mining.

1. INTRODUCTION

Principal Component Analysis (PCA) is a technique used to reduce data dimensionality. It projects data points into the directions of maximal variance within data space. These directions are the eigenvectors of data covariance matrix. In most of the cases, only some few eigenvectors are selected, normally the ones that have the highest eigenvalues. The eigenvalue is equivalent to the variance of a new variable, that is obtained by projecting the data into an eigenvector. The new variables not only have maximal variance, but they are also uncorrelated [Bishop 2006]. PCA is a very well-known technique that is used in several different applications such as face recognition [Turk and Pentland 1991] and text classification [Alencar et al. 2014].

From the perspective of machine learning, PCA is an unsupervised feature extraction technique. Nonetheless, it is also used in supervised tasks such as in classification and regression. Some versions of supervised PCA have been proposed, for example, Barshan et al. [Barshan et al. 2011] proposed a version of supervised PCA for classification. The method defines class representatives and computes PCA for these points. Directions with maximal variances for those points are also the directions that better separate the classes. Another version of supervised PCA was proposed by Bair et al. [Bair et al. 2006] for regression. The technique selects features that have high predictive power and compute PCA using only those features. Therefore, avoiding the interference of features that have high variance but low predictive power.

This work was partially supported by CAPES, CNPq and FACEPE.

The Bayesian approach for classification is very robust and, similarly to PCA, it depends on the data covariance matrix [Duda et al. 2000]. Here, we propose a supervised version of PCA that minimizes the Bayes error rate for classification. The method projects the same features as PCA but selects the ones that minimize the Bayes error rate, while PCA selects the features with maximal variance. Therefore, it can be more suitable for classification task than standard PCA. Since projections of maximal variance might not be the best way to separate data from different classes [Bishop 2006].

The remainder of the article is organized as follows: the next section introduces the mathematical notation and how to use PCA for feature extraction. Section 3 describes the Bayes error rate for classification and the imposed restrictions for calculating it. The proposed method is defined in Section 4. Section 5 presents an analysis, using artificial data, of which features selected by each technique: standard PCA and the proposed method. Section 6 present experiments using real datasets. Conclusion and future work are discussed in Section 7.

2. FEATURE EXTRACTION WITH PCA

Suppose that the dataset is represented in a matrix. The dataset matrix $\mathbf{X}'_{n\times d}$ with n points and d features. Each row of $\mathbf{X}'$ is a data point and each column is a feature.

$$\mathbf{X}' = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}. \tag{1}$$

The j-th point is defined as a d dimensional column vector $\mathbf{x}_j$,

$$\mathbf{x}_j = \begin{bmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jd} \end{bmatrix}, \tag{2}$$

for $j = 1, \ldots, n$ and the data mean vector is

$$\bar{\mathbf{x}} \;=\; \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j. \tag{3}$$

The centered matrix is $\mathbf{X}$ having the j-th row equal to $(\mathbf{x}_j - \bar{\mathbf{x}})^T$:

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ (\mathbf{x}_2 - \bar{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T \end{bmatrix}. \tag{4}$$

The covariance matrix of $\mathbf{X}$ is defined as

$$\mathbf{\Sigma}_{\mathbf{X}} = \frac{1}{n}\mathbf{X}^T\mathbf{X}. \tag{5}$$

Each column $\boldsymbol{\xi}_i$, for $i = 1, \ldots, k$, of the matrix

$$\mathbf{E}_k = [\boldsymbol{\xi}_1 \ldots \boldsymbol{\xi}_k], \tag{6}$$

is an eigenvector of $\mathbf{\Sigma}_{\mathbf{X}}$. $\mathbf{E}_k$ have up to d eigenvectors, for $k = 1, \ldots, d$. Each eigenvector $\boldsymbol{\xi}_i$ have an associated eigenvalue λ_i, which is the variance of the extracted feature $\mathbf{f}_i$,

$$\mathbf{f}_i = \mathbf{X}\boldsymbol{\xi}_i. \tag{7}$$

The value of the i-th extracted feature for the j-th point is w_{ij}, where $\mathbf{f}_i = [w_{1i} \dots w_{ni}]^T$.

The projection of the point $\mathbf{x}_j^T = [x_{j1} \dots x_{jd}]$ for the space of projected features is $\mathbf{w}_j^T = [w_{j1} \dots w_{jk}]$, given by

$$\mathbf{w}_j^T = \mathbf{x}_j^T \mathbf{E}_k. \tag{8}$$

The eigenvectors in $\mathbf{E}_k$ are sorted, so that $\lambda_1 > \ldots > \lambda_k$. In PCA, the points are projected in the directions of maximal variances, these directions are the eigenvectors of the covariance matrix that has the greatest eigenvalues. The new data matrix $\mathbf{W}_{n \times k}$ is defined as:

$$\mathbf{W} = \mathbf{X}\mathbf{E}_k. \tag{9}$$

Each row of this matrix is a point and each column an extracted feature.

The covariance matrix of $\mathbf{W}$ is $\mathbf{\Sigma_W} = n^{-1}\mathbf{W}^T\mathbf{W}$, so that $\mathbf{\Sigma_W} = diag(\lambda_1, \ldots, \lambda_k)$. The variables are uncorrelated since the off-diagonal elements of $\mathbf{\Sigma_W}$ are equal to 0. This property is very important for supervised learning, because it allows the selection of any subset of the projected variables by ignoring their interaction. However, selecting the eigenvectors of highest eigenvalues may not be the best strategy for classification problems, since projections of maximal variance may mix points from different classes [Bishop 2006] within the same region. In Section 5 we show an example which highlights how the component of the highest eigenvalue is less suitable for classification than another component. Therefore, we propose a method of selecting the eigenvectors by minimizing the Bayes error rate for classification.

3. BAYES ERROR RATE

The Bayes error rate for classification is defined as the probability of the classification error, *i.e.*, the expected error rate. This error estimation can have a simplified form by imposing some restrictions. Here, we consider the following five restrictions: (1) The data presents a multivariate normal distribution. (2) The problem has only two classes. (3) The prior probabilities of both classes are equal. (4) Both classes have the same covariance matrix; the same assumption is used for PCA. Finally, (5) the features are statistically independent, similarly to PCA. Then the Bayes error rate is given by [Duda et al. 2000]:

$$P(\text{error}) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{-u^2/2} du. \tag{10}$$

The Bayes error rate decreases as r increases. We define r^2 as the Mahalanobis distance between the mean vectors of the classes ($\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$):

$$r^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \tag{11}$$

$\mathbf{\Sigma}$ is the data covariance matrix, which is the same for both classes; $\boldsymbol{\mu}_1^T = [\mu_{11} \dots \mu_{1d}]$ and $\boldsymbol{\mu}_2^T = [\mu_{21} \dots \mu_{2d}]$ are the mean vector for classes 1 and 2, respectively.

For independent features, the covariance matrix is a diagonal matrix. The off-diagonal elements are the features covariances, which have values equal to zero. This means that each feature is uncorrelated so r has a special form:

$$r = \sqrt{\sum_{i=1}^{d} \left(\frac{\mu_{1i} - \mu_{2i}}{\sigma_i} \right)^2}, \tag{12}$$

where $i = 1, \ldots, d$ are the indexes for the features. The variables μ_{1i} and μ_{2i} are the mean of the feature i for classes 1 and 2, respectively. And σ_i is the variance of the feature i that is the same for both classes.

We emphasize that the probability of classification error decreases as r increases. From Equation (12), we can conclude that each feature contributes for minimizing the probability of classification error. In fact, some feature contributes more than others. The larger the difference between the means of the two classes related to the feature variance, the higher is the contribution of this feature to minimize Bayes error. In the next section, we derive the proposed method based on this criterion of Bayes error minimization.

4. PROPOSED METHOD

Since PCA generates uncorrelated features, and it also considers that the covariance matrix is the same for every class in the dataset (because it computes direction of maximal variance for a covariance matrix estimated for all data), then the Bayes error rate can be minimized proportionally to r, as defined in Equation (12), for features extracted using Equation (9). The proposed method considers these equations to choose the PCA projected variables. However, instead of selecting the directions of maximal variance for the classification task, we select the directions that minimize Bayes error rate.

The problem continues to be restricted to two classes, setting $\mathbf{W} = \mathbf{X}\mathbf{E}_d$, as in Equation (9). However, now the features are extracted for d eigenvectors. We define w_{ij} as the value of the i-th new feature ($i = 1, \ldots, d$) for the j-th point ($j = 1, \ldots, n$). The mean of the i-th feature for the c-th class ($c = 1, 2$) is

$$\bar{w}_{ci} = \frac{\sum_{j=1}^{n} w_{ij}\delta_{jc}}{\sum_{j=1}^{n} \delta_{jc}}, \tag{13}$$

where δ_{jc} is the Dirac's delta function $\delta_{jc} = 1$ if the j-point belongs to the c-th class, and $\delta_{jc} = 0$, otherwise.

According Equation (12), each feature has a relevance for the classification task. Then we propose a score for the relevance of a feature for classification. This score is calculated for each feature extracted with PCA, s_i is the score for the i-th extracted feature:

$$s_i = \begin{cases} (\bar{w}_{1i} - \bar{w}_{2i})^2/\lambda_i & \text{if } \lambda_i \neq 0 \\ 0 & \text{if } \lambda_i = 0 \end{cases}, \tag{14}$$

where λ_i is the eigenvalue of the eigenvector from which the i-th feature were computed, and $\bar{w}_{ci}$ is the mean of the i-th feature for the c-th class ($c = 1, 2$). If $\lambda_i = 0$ the variance of the i-th extracted feature is zero, which means that the variable has the same value for all points. Therefore it is not useful for classification and its score is set to $s_i = 0$. Otherwise, the score is positive and is defined as the absolute value of the difference between the mean of each class divided by the variance of the feature. Features selected according to this score minimize the Bayes error rate. In summary, the proposed method consists of the following steps:

(1) Project the data as $\mathbf{W} = \mathbf{X}\mathbf{E}_d$, similar to Equation (9).
(2) Compute the mean of each feature for each class $\bar{w}_{ci}$, Equation (13).
(3) Compute the score s_i of each feature, Equation (14).
(4) Select k features with the highest score.
(5) Define the projection matrix as:

$$\mathbf{S}_k = [\boldsymbol{\xi}_1 \ldots \boldsymbol{\xi}_k] \tag{15}$$

with the eigenvectors that have the highest scores s_i, such that $s_i \geq s_j$ if $\boldsymbol{\xi}_i \in \mathbf{S}$ and $\boldsymbol{\xi}_j \notin \mathbf{S}$.
(6) Project the data as:

$$\mathbf{V} = \mathbf{X}\mathbf{S}_k, \tag{16}$$

where $\mathbf{V}_{n\times k}$ is the projected data matrix with n points and k discriminant features.

The difference between standard PCA and the proposed method is that the selected features in PCA are the ones of highest eigenvalues (λ_i) and the selected features in the proposed method are the ones with the highest discriminant score (s_i). In the next section, we present an example using an artificial dataset.

5. EXAMPLE WITH ARTIFICIAL DATASET

In this section, we aim to explain how the proposed method works through an example. We use a synthetic dataset to explain better how the proposed method differs from the standard PCA and how it improves the classification accuracy. The dataset has two variables, *i.e.*, each data point has two features. It allows us to visualize the datasets in a plot. The scatter plots of each dataset can be visualized in Figure 1. In these figures, it is also depicted (as a bell curve) the mean and the variance of the normal distribution along each axis. There is a bell curve for each class. Horizontal Axis is the first of the two features, and Vertical Axis is the second feature.

For this dataset, we consider all the five restriction imposed by the proposed method in Section 3 (the data presents a multivariate normal distribution, the problem has only two classes, the prior probabilities of both classes are equal, both classes have the same covariance matrix, and the features are statistically independent). There are 50 examples from each class randomly generate from a normal distribution given a mean vector and a covariance matrix. Half of the examples are randomly chosen as the training set, and the other half is used as the test set for each holdout evaluation. The mean and standard deviation of the accuracy is computed for 100 holdout repetitions. This experimental protocol is the same used with the real datasets (Section 6). We also used the same four classifiers: 1-NN, Naive Bayes, Linear Discriminant, and Decision Tree. For all the four cases, the accuracy of the recognition using raw data (2 original features) and feature extraction (1 extracted feature) with PCA and Proposed method is described in Table I.

For the dataset, the mean vector for class 1 ($\boldsymbol{\mu}_1$), the mean vector for class 2 ($\boldsymbol{\mu}_2$), and shared covariance matrix ($\boldsymbol{\Sigma}$) are defined as:

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}, \; \boldsymbol{\mu}_2 = \begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix}, \; \boldsymbol{\Sigma} = \begin{bmatrix} 1.0 & -0.9 \\ -0.9 & 1.0 \end{bmatrix}.$$

The full dataset can be visualized in Figure 1, it is presented data scatter plot and the distribution of both variables (horizontal and vertical axis). It is easy to note that each feature individually has huge class overlap in the raw data in Figure 1. Figure 2 shows the data transformed by PCA, *i.e.*, all the points of the dataset projected using all the two PCA eigenvectors. After the PCA transformation, the Horizontal Axis is the feature with maximal variance. We highlight that the distribution of this feature has almost the same mean and standard deviation for both classes, *i.e.*, almost a full class overlap. The proposed method gives a higher score for the other feature, the one that presents greater mean separability.

Using the 1-NN classifier the proposed method, with only one feature, has accuracy close to 93%, similar to the raw data (2 features). The PCA (1 feature) has accuracy close to 51%. For the Naive

Table I. The results for the **artificial dataset** showing the Mean Accuracy (M.A.), Standard Deviation (S.D.) and the number of Extracted Features (E.F.) for each method, using 1-NN, Naive Bayes, Decision Tree, and Linear Discriminant classifiers. Maximum mean accuracy within each column is emphasized for each classifier.

Method	E.F.	**Decision Tree**		**Naive Bayes**		**Linear Discrim.**		**1-NN**	
Raw	2	80.3%	(6.0%)	78.5%	(7.3%)	<u>97.2%</u>	(1.8%)	<u>93.0%</u>	(3.4%)
PCA	1	52.1%	(7.4%)	55.0%	(6.5%)	61.0%	(6.6%)	50.9%	(6.8%)
Proposed	1	<u>93.9%</u>	(4.4%)	<u>94.5%</u>	(4.1%)	95.3%	(3.4%)	92.9%	(4.3%)

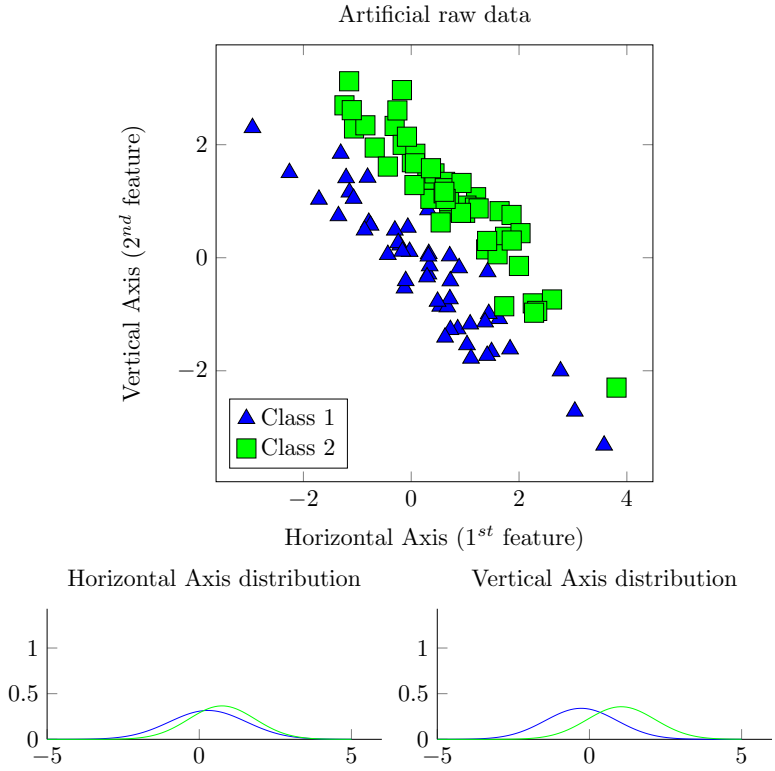

Fig. 1. **Case A.** Scatter plot and the distribution of both variables in **raw data**.

Bayes classifier, the proposed method showed an accuracy close to 95%, which is greater than using raw data (79%). The PCA accuracy is 55%. Using the Linear Discriminant classifier, the proposed method has accuracy close to 95%, raw data 97%, and PCA 61%. For the Decision Tree classifier, the proposed method has accuracy close to 94%, greater than using raw data (80%), while PCA accuracy is close to 52%. In Case A, the proposed method presents accuracy greater than PCA.

In this case, the proposed method has recognition accuracy higher than PCA. The proposed method selects the features that best separate the classes. By the other side, PCA selects the features the spread data the most. The proposed method is the proper method to choose the PCA directions for the classification task. In the next section, we assess the proposed method using nine real datasets.

6. EXPERIMENTS WITH REAL DATASETS

The experiments were performed using nine datasets from the UCI Machine Learning Repository [Lichman 2013]. Each dataset has two classes:

—**Banknote:** the Banknote Authentication Data Set that has 1,372 points and four features.
—**Bank:** the Bank Marketing Data Set that has 4,521 points and 44 features (we converted some of the original 16 categorical features to new binary features).
—**Climate:** the Climate Model Simulation Crashes Data Set that has 540 points and 18 features.
—**Debrecen:** The Diabetic Retinopathy Debrecen Data Set that has 1,151 points and 19 features.
—**Occupancy:** the Occupancy Detection Data Set that has 8,143 points (only the training file) and five features (we remove the date time feature).
—**Pima:** The Pima Indians Diabetes Data Set that has 768 points and eight features.

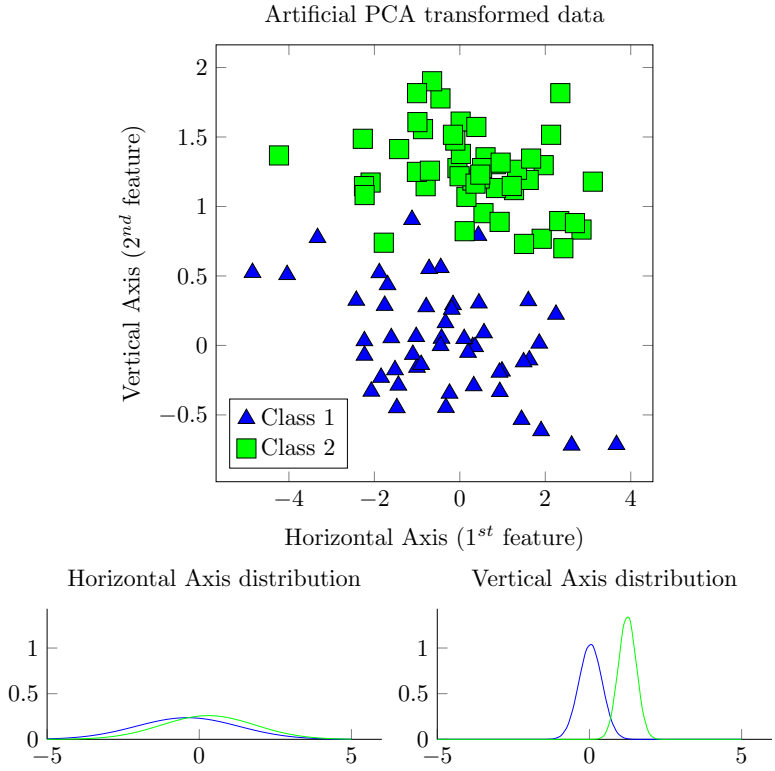

Fig. 2. **Case A.** Scatter plot and the distribution of both variables after **PCA** transformation.

—**Spambase:** the Spambase Data Set that has 4,601 points and 57 features.
—**VColumn:** the Vertebral Column Data Set that has 310 points and six features.
—**WDBC:** the Breast Cancer Wisconsin (Diagnostic) Data Set that has 569 points and 30 features.

Accuracy, the rate of corrected classified points, is the metric used to evaluate the methods. Each mean accuracy is the average accuracy for 100 holdout experiments. In each holdout experiment, 50% of the points from each class were randomly chosen for training and the remaining points were used for testing. The training set was used for both PCA and the proposed method. Both training and test sets were projected using k selected eigenvector, $k = 1, \ldots, d$. The 1-NN (Nearest Neighbor) with Euclidean distance, Naive Bayes with normal kernel smoothing density estimate, pruned Decision Tree with Gini's diversity index and a minimum of 10 nodes per leaf, and Fisher's Linear Discriminant were used for classification. The experiment was performed using Matlab 2017b Statistics and Machine Learning Toolbox.

We calculated the confidence intervals assuming that each mean follows a Student's t distribution. For a 95% confidence level this interval is $[\bar{a} - E,\ \bar{a} + E]$, where $\bar{a}$ is the mean accuracy, $E = 1.984b/\sqrt{100}$, and b is the accuracy standard deviation. If there is no overlap between the confidence intervals of PCA and the proposed method the difference is considered significant [Schenker and Gentleman 2001]. The error bars shown for the datasets in the Figures 3 to 11, represent the confidence intervals. For some figures, the values are too small to appear in the plots. If there is no overlap between the error bars, we consider that the accuracies are significantly different.

The classification accuracy for each dataset and the four classifiers are summarized in Table II and they are discussed in following. For each of the nine datasets, there are 20 mean accuracies in Table II, five average accuracies for each one of the four classifiers. The five results are, from top to bottom:

Table II. Mean accuracy for each dataset and four classifiers. The number of features for the respective accuracy is indicated within square brackets.

		Decision Tree	Naive Bayes	Linear Discriminant	1-NN
Bank	Raw	[48] 87.4%	[48] 88.4%	[48] 90.0%	[48] 84.4%
	Proposed	[1] 83.9%	[3] 89.2%	[9] 89.9%	[1] 82.8%
	PCA	[1] 81.9%	[3] 86.0%	[9] 88.8%	[1] 80.2%
	Proposed	[11] 86.4%	[3] 89.2%	[13] 90.0%	[13] 85.2%
	PCA	[38] 86.2%	[2] 88.5%	[38] 90.1%	[5] 84.5%
Banknote	Raw	[4] 97.5%	[4] 91.4%	[4] 97.6%	[4] 99.9%
	Proposed	[1] 85.7%	[1] 89.0%	[1] 88.9%	[1] 85.0%
	PCA	[1] 69.7%	[1] 69.4%	[1] 61.3%	[1] 68.4%
	Proposed	[4] 98.8%	[4] 97.4%	[4] 97.6%	[4] 99.9%
	PCA	[4] 98.7%	[4] 97.4%	[4] 97.6%	[4] 99.9%
Climate	Raw	[18] 90.4%	[18] 92.0%	[18] 94.5%	[18] 89.0%
	Proposed	[2] 87.7%	[9] 92.2%	[11] 94.0%	[7] 89.9%
	PCA	[2] 86.5%	[9] 91.5%	[11] 92.0%	[7] 86.8%
	Proposed	[7] 88.7%	[11] 92.3%	[18] 94.5%	[10] 90.3%
	PCA	[7] 88.6%	[18] 91.9%	[18] 94.5%	[18] 89.0%
Debrecen	Raw	[19] 61.0%	[19] 55.8%	[19] 71.2%	[19] 61.7%
	Proposed	[3] 63.2%	[1] 62.7%	[3] 68.7%	[3] 62.5%
	PCA	[3] 58.8%	[1] 58.2%	[3] 60.5%	[3] 58.5%
	Proposed	[14] 65.5%	[19] 70.0%	[18] 71.3%	[4] 62.5%
	PCA	[10] 66.2%	[19] 70.0%	[11] 71.8%	[7] 61.8%
Occupancy	Raw	[5] 99.1%	[5] 97.8%	[5] 98.8%	[5] 98.8%
	Proposed	[3] 98.9%	[3] 96.1%	[3] 98.6%	[1] 92.8%
	PCA	[3] 98.9%	[3] 95.8%	[3] 97.6%	[1] 92.8%
	Proposed	[5] 99.0%	[3] 96.1%	[4] 98.8%	[5] 98.8%
	PCA	[4] 99.0%	[5] 96.1%	[5] 98.8%	[4] 98.8%
Pima	Raw	[8] 70.0%	[8] 73.8%	[8] 76.5%	[8] 67.1%
	Proposed	[1] 67.4%	[1] 72.5%	[1] 72.5%	[1] 65.6%
	PCA	[1] 62.6%	[1] 58.7%	[1] 65.4%	[1] 61.4%
	Proposed	[5] 70.1%	[2] 73.5%	[7] 76.5%	[5] 68.5%
	PCA	[6] 70.0%	[8] 72.9%	[8] 76.5%	[5] 67.2%
Spambase	Raw	[57] 90.6%	[57] 56.3%	[57] 88.9%	[57] 79.4%
	Proposed	[3] 82.2%	[3] 82.7%	[4] 80.8%	[2] 76.2%
	PCA	[3] 73.9%	[3] 69.5%	[4] 67.0%	[2] 73.4%
	Proposed	[14] 87.1%	[56] 86.8%	[49] 88.9%	[17] 79.6%
	PCA	[37] 87.0%	[53] 86.9%	[56] 88.9%	[54] 79.5%
VColumn	Raw	[6] 80.2%	[6] 77.0%	[6] 83.0%	[6] 81.5%
	Proposed	[2] 77.2%	[2] 80.9%	[2] 80.4%	[2] 76.4%
	PCA	[2] 73.1%	[2] 75.2%	[2] 71.6%	[2] 72.9%
	Proposed	[4] 80.2%	[3] 82.2%	[3] 83.0%	[6] 81.5%
	PCA	[5] 80.3%	[5] 80.8%	[5] 83.5%	[5] 81.5%
WDBC	Raw	[30] 92.2%	[30] 93.8%	[30] 95.2%	[30] 91.3%
	Proposed	[3] 92.1%	[3] 93.9%	[3] 94.3%	[1] 85.4%
	PCA	[3] 90.2%	[3] 89.6%	[3] 87.5%	[1] 85.4%
	Proposed	[3] 92.1%	[3] 93.9%	[15] 95.3%	[30] 91.3%
	PCA	[5] 91.9%	[5] 92.0%	[17] 95.8%	[5] 91.4%

(1) mean accuracy for raw data,

(2) mean accuracy for the proposed method with maximum difference to PCA,

(3) mean accuracy for the PCA with maximum difference to the proposed method,

(4) maximum mean accuracy for the proposed method, and

(5) maximum mean accuracy for PCA.

Other relevant results that are not described in the table are described in details during the analysis of the results. The next subsections describe the analysis for each dataset.

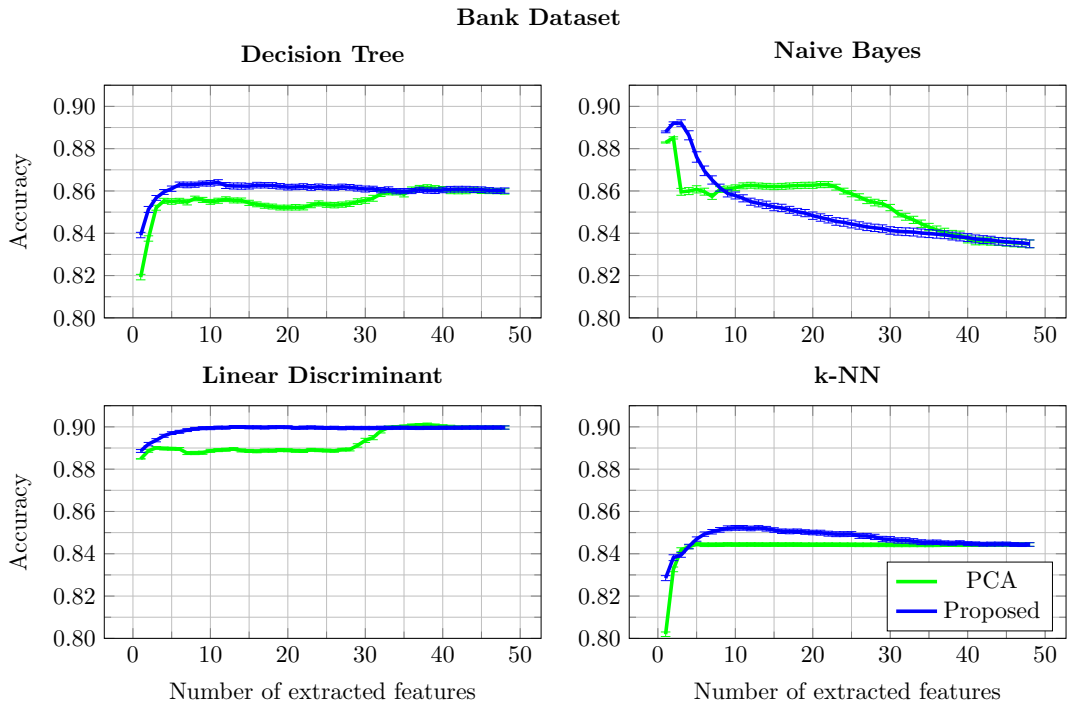

Fig. 3. Mean accuracies for the Bank dataset per number of extracted features, per each classifier.

6.1 Bank Dataset

The maximum accuracy for the Bank dataset is 90% using the Linear Discriminant (LD) classifier. The proposed method (PM) achieve this result using only 13 of the 48 features, and PCA using 34 features. For 38 features with PCA and LD, the mean accuracy is 90.1%, but it is not significantly higher. For the Naive Bayes (NB) classifier using 2 features, the accuracy for PCA is 88.5% and for PM 89.2% (89.21% for three features). For LD with two features, the accuracies are 89.2% (PM) and 88.9% (PCA). For many different numbers of features, considering the four classifiers, the proposed method presented significant higher accuracy, although the difference is only 1% (Figure 3).

6.2 Banknote Dataset

Figure 4 depicts the results for Banknote dataset. The maximum accuracy of 99.9% occurs for the 1-NN classifier using all the four features. However, the proposed method (PM) have an accuracy of 97.5% using only two features, and PCA 85.1%. For Naive Bayes (NB) using four features after projection increases the accuracy to 97.4% comparing to raw data (91.4%), similar fact occurs to the Linear discriminant classifier. The greatest difference is 89.0% (PM) and 69.4% (PCA) for one extracted feature using the NB classifier. A similar difference occurs for other classifiers using the same number of features.

6.3 Climate Dataset

Figure 5 depicts the results for Climate dataset. The maximum accuracy of 94.5% occurs using all the features with Linear Discriminant (LD) classifier. However, the proposed method (PM) has an accuracy of 94% using 11 features. For the same number of features with PCA, the accuracy is 92%. For the Decision Tree classifiers, the accuracy decreases when using more than seven features for both

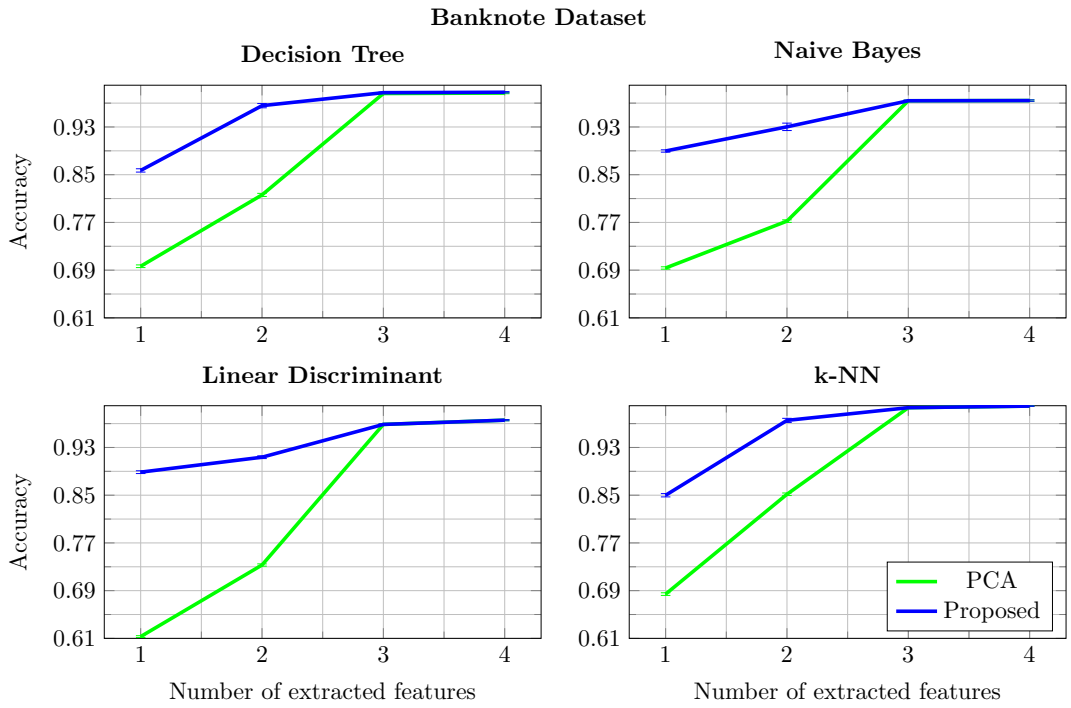

Fig. 4. Mean accuracies for the Banknote dataset per number of extracted features, per each classifier.

methods. PM presents accuracy significantly higher for many numbers of features for all the classifiers. The maximal accuracy for PM is higher or equivalent to PCA in all classifiers, and the PM requires fewer features for Naive Bayes and 1-NN.

6.4 Debrecen Dataset

Figure 6 depicts the results for Debrecen dataset. For this dataset PCA and the proposed method (PM) achieve very similar accuracy using 7 or more extracted features. For Decision Tree (DT) and Linear Discriminant (LD), PCA demands fewer features for the higher accuracy. Except for DT and LD (around ten extract features), PM presents accuracy higher or similar than PCA for the same number of features. Most of the greatest difference occurs for three extracted features; they are 68.7% (PM) and 60.5% (PCA). PM presented a result very close to the maximal accuracy (71.8%, 11 features, PCA) using very few features.

6.5 Occupancy Dataset

Figure 7 depicts the results for Occupancy dataset. For this dataset PCA and the proposed method present very similar accuracies. The greatest difference occurs for three extracted features and Linear Discriminant classifier: 96.1% (proposed) and 95.8% (PCA).

6.6 Pima Dataset

Figure 8 depicts the results for Pima dataset. The greatest difference occurs using a single extracted feature for all the classifiers, for Naive Bayes the results are 72.5% (proposed method) and 58.7% (PCA). The accuracy of the proposed method (PM) with one feature is similar to the accuracy of PCA using all the eight features (72.9%). The greatest of 76.5% accuracy occurs for the Linear Discriminant and PM using seven features.

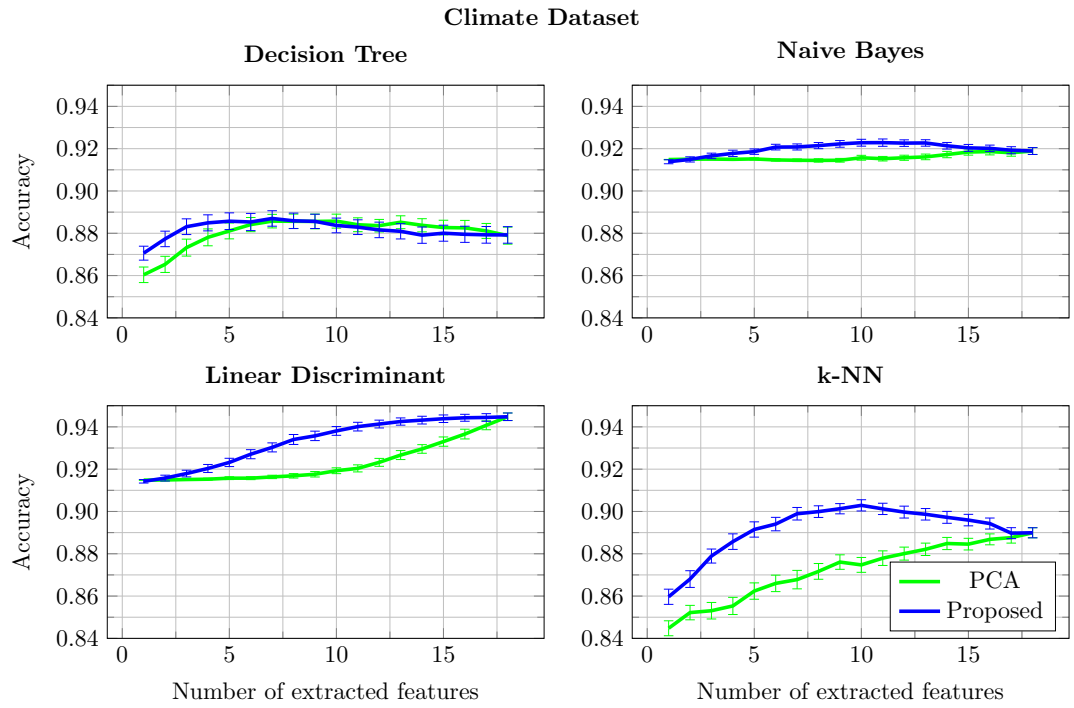

Fig. 5. Mean accuracies for the Climate dataset per number of extracted features, per each classifier.

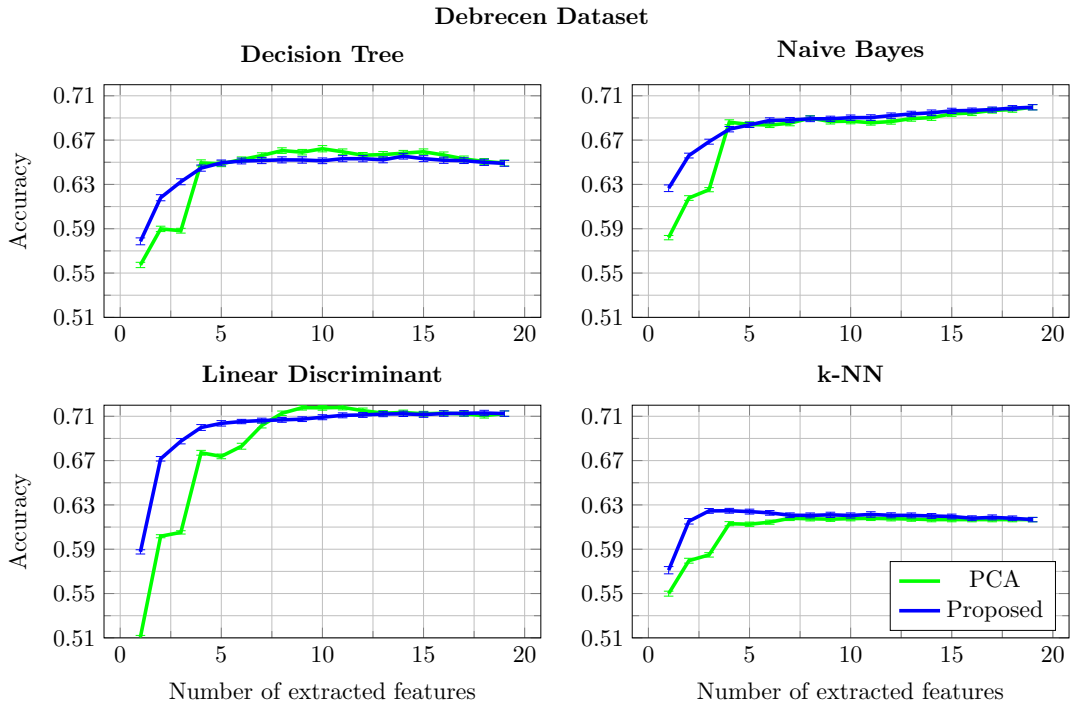

Fig. 6. Mean accuracies for the Debrecen dataset per number of extracted features, per each classifier.

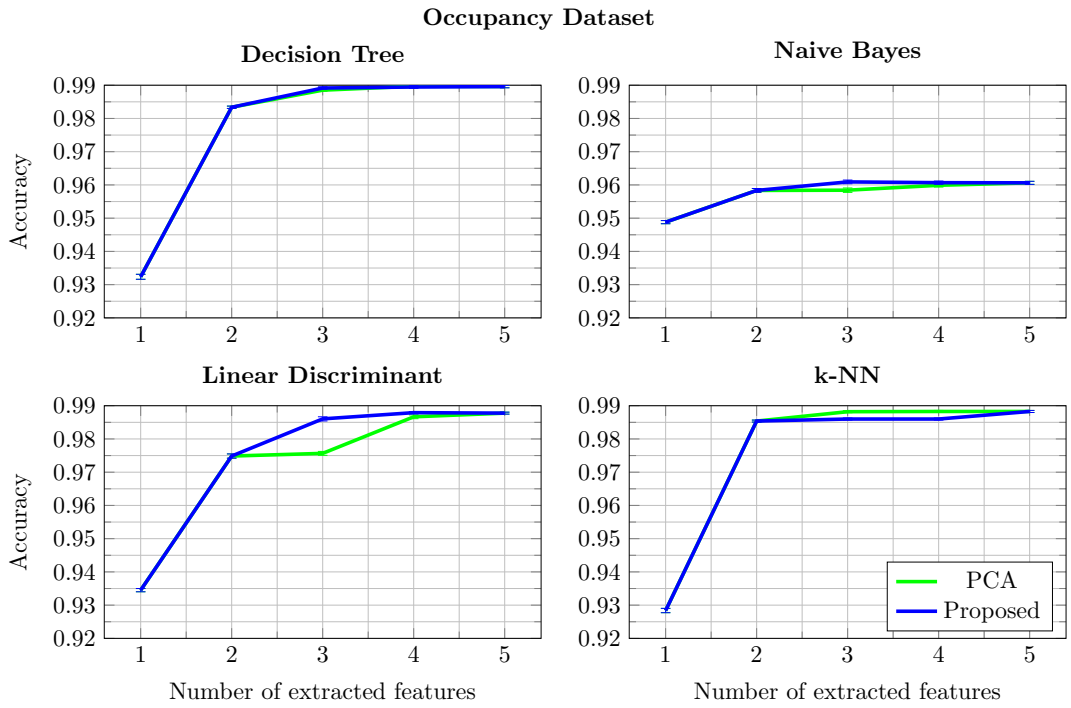

Fig. 7. Mean accuracies for the Occupancy dataset per number of extracted features, per each classifier.

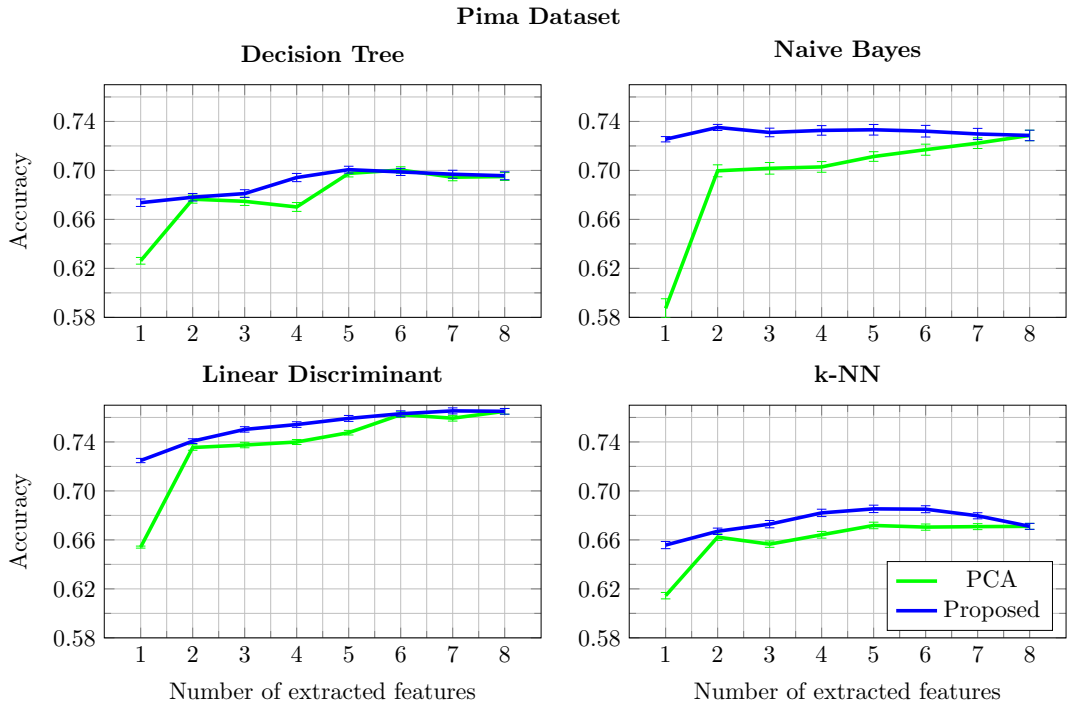

Fig. 8. Mean accuracies for the Pima dataset per number of extracted features, per each classifier.

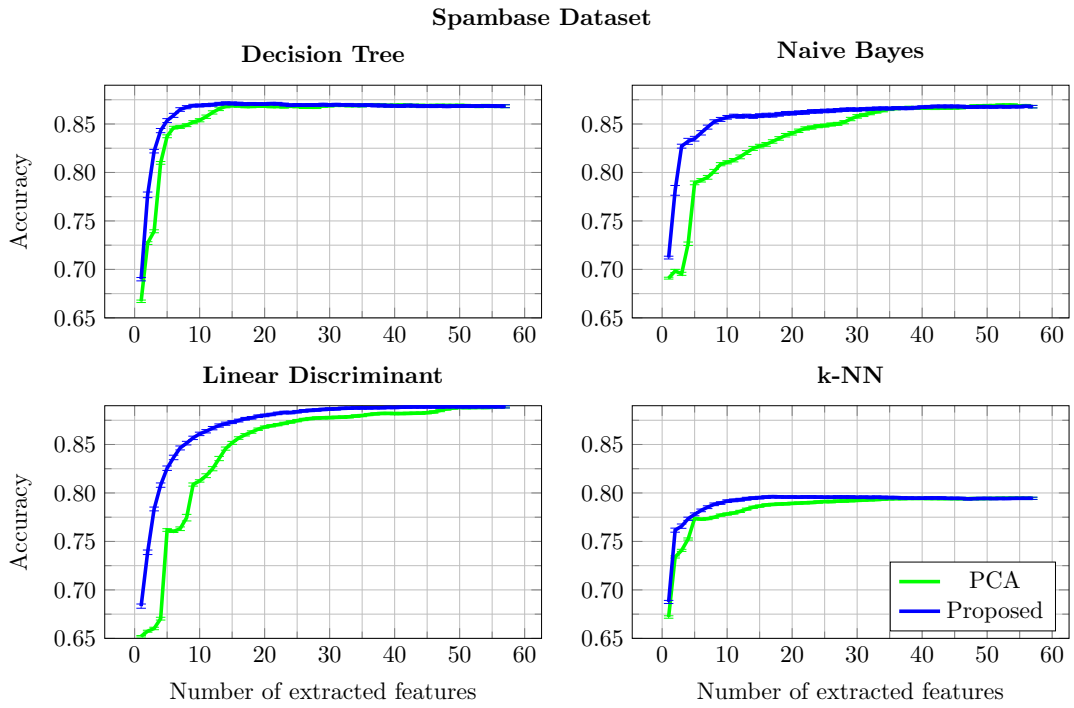

Fig. 9. Mean accuracies for the Spambase dataset per number of extracted features, per each classifier.

6.7 Spambase Dataset

Figure 9 depicts the results for Spambase dataset. PCA and the proposed method (PM) have similar accuracies. The greatest difference occurs for few extracted features. The maximum accuracy of 90.6% occurs for raw data (57 features) and the Decision Tree (DT) classifier. A good trade-off result is for PM with DT 87.1% for 14 extracted features, because it is a high accuracy for few extracted features. PCA has similar accuracy with DT for 37 features (2,64 times more features). The greatest difference between PCA and PM are for Linear Discriminant with four features, 80.8%(PM) and 67% (PCA). A similar difference occurs for Naive Bayes (3 features) 82.7% (PM) and 69.5% (PCA), note that is higher than accuracy for raw data (56.3%).

6.8 VColumn Dataset

Figure 10 depicts the results for VColumn dataset. The greatest difference occurs for the Linear Discriminant (LD) classifier, 80.4% (proposed method) and 71.6% (PCA). The highest accuracy is also achieved using LD, 83.5% (PCA, five features), but it is not significantly different from 83% (PM, three features). For 2 and three extracted features PM has accuracy significantly higher, and close to the maximum, with every classifier.

6.9 WDBC Dataset

Figure 11 depicts the results for WDBC dataset. The greatest accuracy occurs for the Linear Discriminant (LD) classifiers, 95.8% (PCA, 17 features), 95.3% (proposed method, 15 features), and 95.2% (raw data, 30 features). The greatest difference also occurs for LD (3 features), 94.3% (proposed) and 87.5% (PCA), almost 7%. The proposed method presented accuracy very close to the maximum using only 10% (3 out 30) of the features.

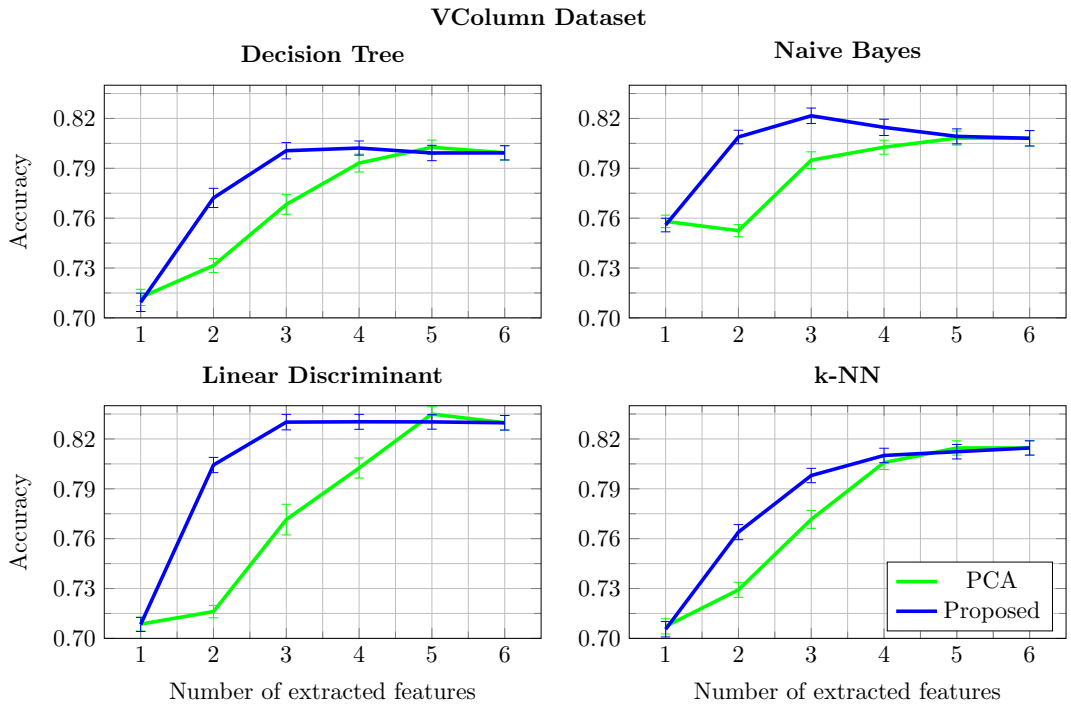

Fig. 10. Mean accuracies for the VColumn dataset per number of extracted features, per each classifier.

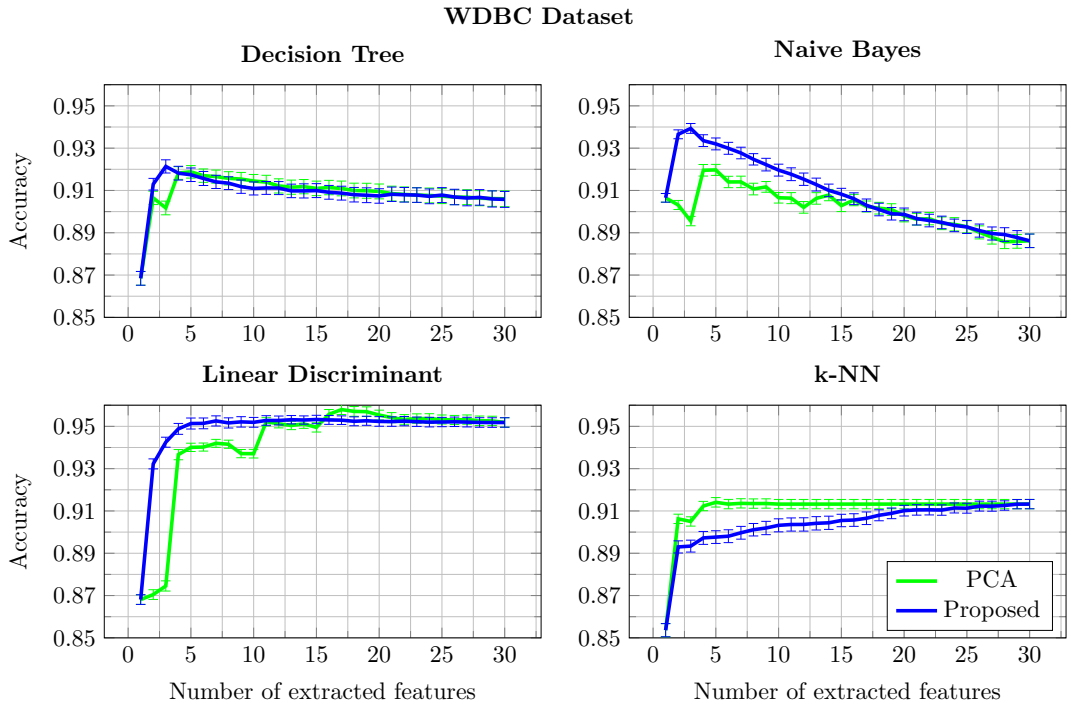

Fig. 11. Mean accuracies for the WDBC dataset per number of extracted features, per each classifier

6.10 Final remarks

For all the 36 (4 classifiers × 9) plots, the only case where PCA has accuracy higher than proposed method (PM) for few extracted features is for 1-NN and WDBC dataset. For the Bank dataset with Naive Bayes classifier, there is a case where PCA accuracy is significantly greater than proposed method (from 10 to 34 extracted features), but those accuracies are much smaller than the accuracy for two extracted features. In 34 out 36 cases, PM present accuracy close or significantly greater than PCA for the same number of extracted features. The greatest accuracy differences occur using less than half of the features. The proposed method can extract more discriminant accuracy and reduce accuracy, and it seems more suitable for dimensionality reduction in supervised tasks.

7. CONCLUSION

We proposed a feature extraction technique that is similar to PCA but selects features that minimize the Bayes error rate instead of features that maximizes the variance. The method presented a higher mean accuracy compared to PCA in two real datasets using a small number of features. The accuracy was evaluated using four distinct classifiers. Experiments with four artificial datasets show how the proposed method chooses discriminant features. We present some graphical examples to describe how the proposed method selects the directions that reduce the overlap between the classes.

For future work, the proposed method can be extended to problems with more than two classes. Also, the Bayes error rate can be computed using fewer restriction. By doing these extensions, it is possible to evaluate the proposed technique using other real datasets. Another research investigation is to test the method presented herein other PCA-based techniques such as Fractional Eigenfaces [de Carvalho et al. 2014] and Supervised Fractional Eigenfaces [de Carvalho et al. 2015].

REFERENCES

Alencar, A. S. C., Gomes, J. P. P., Souza, A. H., Freire, L. A. M., Silva, J. W. F., Andrade, R. M. C., and Castro, M. F. Regularized supervised distance preserving projections for short-text classification. In *Brazilian Conference on Intelligent Systems (BRACIS)*. Sao Carlos, Brazil, pp. 216–221, 2014.

Bair, E., Hastie, T., Paul, D., and Tibshirani, R. Prediction by supervised principal components. *Journal of the American Statistical Association* 101 (473): 119–137, 2006.

Barshan, E., Ghodsi, A., Azimifar, Z., and Jahromi, M. Z. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition* 44 (7): 1357 – 1371, 2011.

Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

de Carvalho, T. B. A., Costa, A. M., Sibaldo, M. A. A., Tsang, I. R., and Cavalcanti, G. D. C. Supervised fractional eigenfaces. In *IEEE International Conference on Image Processing (ICIP)*. Quebec, Canada, pp. 552–555, 2015.

de Carvalho, T. B. A., Sibaldo, M. A. A., Tsang, I. R., Cavalcanti, G. D. C., Tsang, I. J., and Sijbers, J. Fractional eigenfaces. In *IEEE International Conference on Image Processing (ICIP)*. Paris, France, pp. 258–262, 2014.

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

Lichman, M. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.

Schenker, N. and Gentleman, J. F. Statistical practice: On judging the significance of differences by examining the overlap between confidence intervals. *The American Statistician* 55 (3): 182–186, Aug., 2001.

Turk, M. and Pentland, A. Eigenfaces for recognition. *J. Cognitive Neuroscience* 3 (1): 71–86, 1991.