SVM-based Supervised and Unsupervised Classification Schemes

LUMINITA STATE
University of Pitesti
Faculty of Mathematics and Computer Science
1 Targu din Vale St., Pitesti 110040
ROMANIA
lstate@clicknet.ro

IULIANA PARASCHIV-MUNTEANU
University of Bucharest
Faculty of Mathematics and Computer Science
14 Academiei St., Bucharest 010014
ROMANIA
pmiulia@fmi.unibuc.ro

Abstract: The aim of the research reported is to propose a training algorithm for support vector machine based on kernel functions and to test its performance in case of non-linearly separable data. The training is based on the Sequential Minimal Optimization introduced by J.C. Platt in 1999. Several classifications schemes resulted by combining the SVM and the 2-means methods are proposed in the fifth section of the paper. A series of conclusions derived experimentally concerning the comparative analysis of the performances proved by the proposed methods are summarized in the final part of the paper. The tests were performed on samples randomly generated from Gaussian two-dimensional distributions, and on data available in Wisconsin Diagnostic Breast Cancer Database.

Key–Words: Support Vector Machine, Pattern recognition, Statistical learning theory, Kernel functions, Principal Components Analysis, k-means Algorithm.

1 Introduction

In empirical data modeling a process of induction is used to build up a model of the system, from which it is hoped to deduce responses of the system that have yet to be observed. Ultimately the quantity and quality of the observations govern the performance of this empirical model.

The Support Vector Machines (SVM) is a pattern classification technique developed by Vladimir Vapnik and his team at AT&T Bell Laboratories as an alternative training technique for Polynomial, Radial Basis Function and Multi-Layer Perceptron classifiers in which the parameters are determined by solving a Quadratic Programming (QP) Problem with linear inequality and equality constraints, the number of variables in the QP problem being equal to the size of learning data.

The learning problem setting for SVMs is as follows: there is some unknown and nonlinear dependency (mapping, function) $y = f(x)$ between some high-dimensional input vector x and scalar output y (or the vector output y as in the case of multiclass SVMs). There is no information about the underlying joint probability functions. Thus, one must perform a distribution-free learning. The SVMs belong to the supervised learning techniques because the only information available is a finite training data set $\mathcal{S}$ consisting of labeled examples (x_i, y_i) where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.

In training SVM the decision boundaries are determined directly from the training data so that the separating margins of decision boundaries are maximized in the high-dimensional space called feature space. This learning strategy, based on statistical learning theory developed by Vapnik ([6],[22]), minimizes the classification errors of the training data and the unknown data.

The paper aims to present the results of the research toward improving the performance expressed in accuracy and time complexity of SVM implementations. The proposed supervised training algorithm for SVM is essentially based on kernel functions of polynomial and exponential types. The implementation of the search process for soft margin hyperplane uses a slight modification of Sequential Minimal Optimization (SMO) algorithm introduced by Platt ([17]) in 1999, to solve the quadratic programming problem involved in the learning process. The SMO is a simple algorithm that quickly solves the SVM problem by decomposing the overall quadratic programming problem into smaller quadratic programming sub-problems without any extra matrix storage and without invoking an iterative numerical routine for each sub-problem. The overall memory requirements of the SMO algorithm is linear in the size of training data and therefore it allows the use of large size samples in the training process. The proposed method was tested on simulated data and on medical data freely offered by Wisconsin Diagnostic Breast Cancer Database, UCI Machine Learning Repository: Data Sets, http://archive.ics.uci.edu/ml/datasets.html. We

propose a combined methodology resulted by using Principal Component Analysis (PCA) means for dimensionality reduction in the space of initial data representations with kernel-based SVM to allow possible inseparability. We also propose classification schemes that combine kernel-based SVM with k-means as a method of unsupervised learning. Using a combination of both supervised and unsupervised learning methods yielded to very good experimental results presented in the fifth section of the paper.

2 General Presentation of Support Vector Machine-based Classification

Support Vector Machine is a relatively new model-free paradigm in Machine Learning. Briefly, SVM is a supervised learning technique of parametric type, the parameters of the discrimination function being learned directly from data. Let us consider a two-class classification problem the unique information concerning classes being represented by a finite sequence of labeled data,

$$\mathcal{S}=\left\{ (x_i,y_i)\mid x_i=\left(x_i^{(1)},\ldots,x_i^{(d)}\right)^T\in \mathbb{R}^d, \; y_i\in\{-1,1\},\; i=\overline{1,N}\right\}. \quad (1)$$

The first component of each pair (x_i, y_i) of $\mathcal{S}$ represents an instance coming from the class of label y_i.

2.1 The case of linearly separable data

The sequence is linearly separable if there exists a linear discriminant function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ that separates the examples of $\mathcal{S}$, that is

$$f(x) = b + w_1x^{(1)} + \ldots + w_dx^{(d)}, \quad (2)$$

for each $x=\left(x^{(1)},\ldots,x^{(d)}\right)^T\in\mathbb{R}^d$, such that for any $(x_i, y_i) \in \mathcal{S}$, $f(x_i) > 0$ if $y_i = 1$, and $f(x_i) < 0$ if $y_i = 1$.

The linear separability can be expressed by the existence of the parameters $b \in \mathbb{R}$ and $w \in \mathbb{R}^d$ such that $w^Tz_i + b > 0$ where $z_i = y_ix_i$ and $w = (w_1,\ldots,w_d)^T$. In such a case we say the hyperplane

$$H_{w,b} \; : \quad w^Tx + b = 0. \quad (3)$$

separates without errors $\mathcal{S}$.

In a SVM-based approach the search for a solution yields to the constrained quadratic optimization problem

$$\begin{cases} \text{minimize } \Phi(w) \\ y_i\left(w^Tx_i+b\right)\geq 1, \quad i=\overline{1,N}, \end{cases} \quad (4)$$

where $\Phi(w)=\frac{1}{2}\|w\|^2$.

Using the Lagrange multipliers method the solution of the optimization problem (4) is given by the primal-dual solution of the optimization problem on the objective function

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^N\alpha_i\left(y_i\left((w^Tx_i+b\right)-1\right), \quad (5)$$

where $\alpha_1,\ldots,\alpha_N$ are the Lagrange multiplies.

Using standard arguments ([1]) the problem reduces to the simpler optimization problem

$$\begin{cases} \max\left(-\frac{1}{2}\sum_{i=1}^N\sum_{k=1}^N\left(\alpha_i\alpha_ky_iy_k\left(x_i^Tx_k\right)\right)+\sum_{i=1}^N\alpha_i\right) \\ \alpha_i\geq 0, \quad \forall\, 1\leq i\leq N, \\ \sum_{i=1}^N\alpha_iy_i=0. \end{cases} \quad (6)$$

If $\alpha^*=(\alpha_1^*,\ldots,\alpha_N^*)$ is a solution of (6), we say that x_i is a *support vector* if $\alpha_i^*\neq 0$. If $\mathcal{S}_1$ is the set of support vector then the optimal solution of (4) is

$$\begin{aligned} w^* &= \sum_{i=1}^N\alpha_i^*y_ix_i, \\ b^* &= \frac{1}{|\mathcal{S}_1|}\sum_{x_i\in\mathcal{S}_1}\left(y_i-\sum_{x_j\in\mathcal{S}_1}\alpha_j^*y_j\left(x_i^Tx_j\right)\right), \end{aligned} \quad (7)$$

that is the discrimination function is

$$f^*(x)=(w^*)^Tx+b^*=\sum_{i=1}^N\alpha_i^*y_i\left(x_i^Tx\right)+b^*.$$

The computation involved in solving the optimization problem (6) can be carried out using the algorithm $SVM1$ ([20], 2009). Briefly, the algorithm SVM1 works as follows:

Algorithm $SVM1$ ([16])

Input: $\mathcal{S}=\left\{(x_i,y_i)\mid x_i\in\mathbf{R}^d, y_i\in\{-1,1\}, i=\overline{1,N}\right\}$

Step 1. Compute the matrix $D=(d_{ik})$ *of entries*

$$d_{ik}=y_iy_k\left(x_i\right)^Tx_k,\; i,k=\overline{1,N};$$

Step 2. Solve the constrained optimization problem

$$\begin{cases} \alpha^*=arg\left(\max_{\alpha\in\mathbf{R}^N}\left(\alpha^T\mathbf{1}-\frac{1}{2}\alpha^TD\alpha\right)\right), \\ \alpha_i\geq 0, \quad \forall\, 1\leq i\leq N, \\ \sum_{i=1}^N\alpha_iy_i=0, \end{cases}$$

Step 3. Select two support vectors x_r, x_s such that

$$\alpha_r^* > 0\,,\ \alpha_s^* > 0\,,\ y_r = -1\,,\ y_s = 1.$$

Step 4. Compute the parameters w^*, b^* of the optimal separating hyperplane,

$$\begin{cases} w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i\,, \\ b^* = -\frac{1}{2}\left(w^*\right)^T (x_r + x_s) \end{cases}$$

and the width of the separating area

$$\rho\left(w^*, b^*\right) = \frac{2}{\|w^*\|}$$

Output: $w^*, b^*, \rho\left(w^*, b^*\right)$.

2.2 The general case

In real life situations the tests to check whether the data are linear separable are costly from computational point of view. Moreover, even in cases when a test for linear separability is used, frequently enough we find out that the labeled data are not linear separable. The method presented in 2.1 was generalized yielding to the concept of soft margin hyperplane by Cortes and Vapnik ([6]). The method is essentially a regularization technique that uses the term expressing the effect of classification errors set

$$\Phi_\sigma\left(\xi_1, \ldots, \xi_N\right) = \sum_{i=1}^{N} \xi_i^\sigma\,, \tag{8}$$

where σ is a positive constant and the non-negative slack variables $\xi_i\,, 1 \le i \le N$, are introduced to allow inseparability. The soft margin hyperplane is given by a solution of the constrained optimization problem

$$\begin{cases} \text{minimize}\left(\frac{1}{2}\|w\|^2 + c\,F\left(\sum_{i=1}^{N} \xi_i^\sigma\right)\right) \\ y_i\left(w^T x_i + b\right) \ge 1 - \xi_i\,,\ \forall\, 1 \le i \le N\,, \\ \xi_i \ge 0\,,\ \forall\, 1 \le i \le N\,, \end{cases} \tag{9}$$

where c is a given positive constant and F is a monotone increasing convex function such that $F(0) = 0$ holds. By applying the Lagrange multipliers method in the particular case when $F(u) = u^k$ and $\sigma = 1$, we obtain the objective function

$$\begin{aligned} L(w\,, \xi\,, b\,, \alpha\,, \beta) = \ & \frac{1}{2}\|w\|^2 + c\left(\sum_{i=1}^{N} \xi_i\right)^k - \\ & \sum_{i=1}^{N} \alpha_i\left(y_i\left(w^T x_i + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{N} \beta_i \xi_i\,, \end{aligned} \tag{10}$$

where $\alpha = (\alpha_1, \ldots, \alpha_N)$ and $\beta = (\beta_1, \ldots, \beta_N)$ are the Lagrange multipliers.

Therefore, in order to solve the constrained problem (9) the objective function L should be minimized with respect to w, $\xi = (\xi_1, \ldots, \xi_N)$ and b and maximized with respect to the non-negative parameters $\alpha_i \ge 0$ and $\beta_i \ge 0$, $1 \le i \le N$. Using standard arguments we get

$$\begin{aligned} & \left.\frac{\partial L}{\partial w}\right|_{w=w^*} = w^* - \sum_{i=1}^{N} \alpha_i y_i x_i = 0\,, \\ & \left.\frac{\partial L}{\partial} b\right|_{b=b^*} = \sum_{i=1}^{N} \alpha_i y_i = 0\,, \end{aligned} \tag{11}$$

$$\left.\frac{\partial L}{\partial \xi_i}\right|_{\xi_i=\xi_i^*} = k\,c\left(\sum_{i=1}^{N} \xi_i^*\right)^{k-1} - \alpha_i - \beta_i = 0\,,\ i = \overline{1, N}. \tag{12}$$

In the following we will refer to the particular case $k = 1$. In this case we get

$$\begin{aligned} & w^* = \sum_{i=1}^{N} \alpha_i y_i x_i\,, \\ & \sum_{i=1}^{N} \alpha_i y_i = 0\,, \\ & c = \alpha_i + \beta_i\,, \quad 1 \le i \le N\,. \end{aligned} \tag{13}$$

A hyperplane that assures the minimum number of errors in separating the data is given by the solution of the constrained optimization problem

$$\begin{cases} \max\limits_{\alpha}\left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j\right), \\ \sum_{i=1}^{N} \alpha_i y_i = 0\,, \quad 0 \le \alpha_i \le c\,,\ i = \overline{1, N}. \end{cases} \tag{14}$$

The computation of soft margin hyperplane is carried out by the algorithm SVM2 ([20], 2009)

Algorithm $SVM2$ ([16])

Input: $\mathcal{S} = \left\{(x_i, y_i) \,|\, x_i \in \mathbf{R}^n, y_i \in \{-1, 1\}, i = \overline{1, N}\right\}$

$c \in (0, \infty)$

Step 1. Compute the matrix $D = (d_{ik})$ of entries

$$d_{ik} = y_i y_k \left(x_i\right)^T x_k\,,\ i, k = \overline{1, N}\,;$$

Step 2. Solve the constrained optimization problem

$$\begin{cases} \alpha^* = arg\left(\max\limits_{\alpha \in \mathbf{R}^N}\left(\alpha^T \mathbf{1} - \frac{1}{2}\,\alpha^T D \alpha - \frac{(\alpha_{max})^2}{4\,c}\right)\right), \\ \alpha_i \ge 0\,, \quad \forall\, 1 \le i \le N\,, \\ \sum_{i=1}^{N} \alpha_i y_i = 0\,, \end{cases}$$

where $\alpha_{max} = \max\{\alpha_1, \ldots, \alpha_N\}$

Step 3. Select two support vectors x_r, x_s,

$$\alpha_r^* > 0\,,\ \alpha_s^* > 0\,,\ y_r = -1\,,\ y_s = 1.$$

Step 4. Compute the parameters w^*, b^* *of the soft margin hyperplane,*

$$\begin{cases} w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i\,, \\ b^* = -\frac{1}{2}\left(w^*\right)^T (x_r + x_s) \end{cases}$$

and the width of the separating area

$$\rho\left(w^*, b^*\right) = \frac{2}{\|w^*\|}$$

Output: $w^*, b^*, \rho\left(w^*, b^*\right)$.

2.3 The kernel trick

In cases when the data are strong non-separable and the performance corresponding to the soft margin hyperplane is poor the data are projected on a higher dimensional space $\mathbb{R}^m$, $m > d$, using a non-linear mapping function $g : \mathbb{R}^d \longrightarrow \mathbb{R}^m$. The explicit functional expression of the mapping function g is 'hidden' by the kernel trick. A symmetric function $K : R^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ is a positive semi-definite kernel ([14], [15]) if for any positive natural number M and for any $(h_1, \ldots, h_M) \in \mathbb{R}^M$, $\{x_1, \ldots, x_M\} \subset \mathbb{R}^d$

$$\sum_{i,j=1}^{M} h_i h_j K\left(x_i, x_j\right) \geq 0\,. \tag{15}$$

According to the Mercer theorem ([14], [15]) if K is a positive semi-definite kernel then there exists a mapping function $g : \mathbb{R}^d \longrightarrow \mathbb{R}^m$ such that

$$K\left(x, x'\right) = \left(g(x)\right)^T g\left(x'\right)\,, \quad \forall\, x, x' \in \mathbb{R}^d\,. \tag{16}$$

The kernel trick in learning soft margin hyperplane is twofold. On one hand, selecting a positive semi-defined kernel the computation of the soft margin hyperplane is performed in a higher dimensional space without using the explicit function expression of the mapping function g. On the other hand, the computation in the higher dimensional space is in fact carried out in terms of the computations in the initial space, that is it can be carried out without involving increased computational complexity. In our tests we used two types of kernels namely polynomials

$$K\left(x, x'\right) = \left(x^T x' + 1\right)^r\,, \tag{17}$$

and exponentials, respectively

$$K\left(x, x'\right) = \exp\left(-\gamma \left\|x - x'\right\|^2\right)\,, \quad \gamma > 0\,. \tag{18}$$

In order to determine the soft margin hyperplane in the higher dimensional space $\mathbb{R}^m$ we have to solve the constrained optimization problem

$$\begin{cases} \min\limits_{w \in \mathbb{R}^m,\, b \in \mathbb{R}} L(w, b, \xi) \\ \\ y_i\left(w^T g\left(x_i\right) + b\right) \geq 1 - \xi_i\,, \quad i = \overline{1, N} \\ \xi_i \geq 0\,, \quad i = \overline{1, N}\,, \end{cases} \tag{19}$$

where $L(w, b, \xi) = \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{N} \xi_i$ and g is the mapping function. Using the Lagrange multiplies method, the dual problem resulted from the Karush-Kuhn-Tucker conditions is

$$\begin{cases} \max\limits_{\alpha \in \mathbb{R}^N} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right) \right) \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c\,, \quad i = \overline{1, N}\,, \end{cases} \tag{20}$$

If $\alpha^* = (\alpha_1^*, \ldots, \alpha_N^*)$ is a solution of (20) and $\mathcal{S}_1$ is the set of support vectors then

$$b^* = \frac{1}{|\mathcal{S}_1|} \sum_{x_i \in \mathcal{S}_1} \left(y_i - \sum_{x_j \in \mathcal{S}_1} \alpha_j^* y_j K\left(x_i, x_j\right) \right)\,. \tag{21}$$

and the discriminant function is

$$f(x) = \operatorname{sgn}\left(\sum_{i \in \mathcal{S}_1} \alpha_i^* y_i\, K\left(x_i\,, x\right) + b^* \right)\,. \tag{22}$$

Consequently, $\|w^*\| = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i^* \alpha_j^* y_i y_j K\left(x_i\,, x_j\right)}$ and the width of the separating area is $\rho = \frac{2}{\|w^*\|}$.

2.4 The SMO Algorithm for Solving the Dual Problem

The Sequential Minimal Optimization (SMO) algorithm was introduced by Platt ([17]) as an iterative method for solving constrained optimization problem of the type

$$\begin{cases} \min\limits_{\alpha \in \mathbb{R}^N} W(\alpha) \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c\,, \quad i = \overline{1, N}\,. \end{cases} \tag{23}$$

In our tests we applied the SMO algorithm to minimize the function

$$W(\alpha) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) .$$

The SMO is a simple algorithm that quickly solves the SVM problem by decomposing the overall quadratic programming problem into smaller quadratic programming sub-problems without any extra matrix storage and without invoking an iterative numerical routine for each sub-problem. SMO algorithm chooses to solve the smallest possible optimization problem at every step, involving only two Lagrange multipliers because a linear equality constraint has to hold for the multipliers. At every step, SMO chooses two Lagrange multipliers to jointly optimize, finds the optimal values for these multipliers and updates the SVM to reflect the new optimal values.

3 Unsupervised learning (clustering) using the k-means method

Center-based clustering algorithms are very efficient for clustering large databases and high-dimensional databases. They have own objective functions which define how good a clustering solution is, the goal being to minimize the objective function. Clusters found by center-based algorithms have convex shapes and each cluster is represented by a center. The k-means algorithm (MacQueen [13], 1967) was designed to cluster numerical data, each cluster having a center called the *mean*.

Let $\mathcal{D} = \{x_1, \ldots, x_N\} \subset \mathbf{R}^d$ be the data set, k a given positive integer, and $\mathcal{C}_1, \ldots, \mathcal{C}_k$ pairwise disjoint clusters of $\mathcal{D}$, that is, $\bigcup_{i=1}^{k} \mathcal{C}_i = \mathcal{D}$, $\mathcal{C}_i \cap \mathcal{C}_j$, $\forall\, i \neq j$. If we denote by $\mu(\mathcal{C}_i)$ the center of $\mathcal{C}_i$ then *the inertia momentum (error)* is expressed by

$$\varepsilon = \sum_{i=1}^{k} \sum_{x \in \mathcal{C}_i} d^2(x, \mu(\mathcal{C}_i)) , \tag{24}$$

where d is a convenable distance function on $\mathbf{R}^d$. In the following we take d as being the Euclidean distance on $\mathbf{R}^d$, $d(x, y) = \|x - y\|$.

The k-means methods proceeds, for a given initial k clusters, by allocating the remaining data to the nearest clusters and then repeatedly changing the membership of the clusters according to the error function until the error function does not change significantly or the membership of the clusters no longer changes.

The k-means algorithm can be treated as an optimization problem where the goal is to minimize a given objective function under certain constrains.

We denote by $\mathcal{C}$ the set of all subsets of $\mathbf{R}^d$ of cardinal k; any particular $Q = \{q_1, \ldots, q_k\} \in \mathcal{C}$ is called a set of possible centers.

A system of k pairwise disjoint clusters of $\mathcal{D}$ can be obviously represented in terms a matrix $W = (w_{il}) \in \mathcal{M}_{N \times k}(\mathbf{R})$ such that

$$\begin{aligned} &(i) \quad w_{il} \in \{0, 1\}, \quad i = \overline{1, N}, \; l = \overline{1, k} \\ &(ii) \quad \sum_{l=1}^{k} w_{ik} = 1, \quad i = \overline{1, N}. \end{aligned} \tag{25}$$

The k-means algorithm can be formulated as the constrained optimization problem:

$$\begin{cases} \min\limits_{W \in \mathcal{M}_{N \times k}(\mathbf{R}),\, Q \in \mathcal{C}} P(W, Q) \\ w_{il} \in \{0, 1\}, \quad i = \overline{1, N}, \; l = \overline{1, k}, \\ \sum\limits_{l=1}^{k} w_{ik} = 1, \quad i = \overline{1, N}, \end{cases} \tag{26}$$

where the objective function is defined as

$$P(W, Q) = \sum_{i=1}^{N} \sum_{l=1}^{k} w_{il} \|x_i - q_l\|^2 . \tag{27}$$

The problem (10) can be solved by decomposing it into two simple problems P_1 and P_2 and iteratively solving them, where

P_1. Fix $Q = \widehat{Q} \in \mathcal{C}$ and solve the reduced constrained optimization problem for $P\left(W, \widehat{Q}\right)$.

P_2. Fix $W = \widehat{W} \in \mathcal{M}_{N \times k}(\mathbf{R})$ and solve the reduced unconstrained optimization problem for $P\left(\widehat{W}, Q\right)$.

The solutions of these problems are given by the following theorems:

Theorem 1 *For any fixed $\widehat{Q} = \{\widehat{q}_1, \ldots, \widehat{q}_k\}$ a set of centers, the function $P\left(W, \widehat{Q}\right)$ is minimized if and only if W satisfies the conditions*

$$\begin{aligned} &w_{il} = 0 \Longleftrightarrow \|x_i - \widehat{q}_l\| > \min_{1 \le t \le k} \|x_i - \widehat{q}_t\| , \\ &w_{il} = 1 \Longrightarrow \|x_i - \widehat{q}_l\| = \min_{1 \le t \le k} \|x_i - \widehat{q}_t\| , \\ &\sum_{j=1}^{k} w_{ij} = 1 , \end{aligned}$$

for any $i = \overline{1, N}$, $l = \overline{1, k}$.

Proof: Let $W^{(0)}=\left(w_{il}^{(0)}\right)$ where

$$w_{il}^{(0)}=\begin{cases}1\,,\ \|x_i-\widehat{q}_l\|=\min\limits_{1\le t\le k}\|x_i-\widehat{q}_t\|\\ 0\,,\ \text{otherwise}\end{cases},$$

$\sum\limits_{i=1}^{N} w_{il}^{(0)}=1,\ i=\overline{1,N}.$

Then for any $W\in\mathcal{M}_{N\times k}(\mathbf{R})$ satisfying the constraints of (10), if we denote by l_i the index such that that $w_{il_i}=1$ and $w_{ij}=0$ for $j\neq l_i$, we get

$$P\left(W,\widehat{Q}\right)=\|x_1-\widehat{q}_{l_1}\|^2+\cdots+\|x_N-\widehat{q}_{l_N}\|^2\ge \min_{1\le t\le k}\|x_1-\widehat{q}_t\|^2+\cdots+\min_{1\le t\le k}\|x_N-\widehat{q}_t\|^2= P\left(W^{(0)},\widehat{Q}\right),$$

that is $W^{(0)}$ is a solution of P_1.

Let W a solution of P_1. If there exists i such that $w_{il}=1$ and $\|x_i-q_{l_i}\|>\min\limits_{1\le t\le k}\|x_i-q_t\|=\|x_i-q_{l_0}\|$ then for $W'=\left(w'_{il}\right)$ where $w'_{jl}=w_{jl}$ for $j\neq i$ and $w'_{il}=\begin{cases}1\,,\ l=l_0\\ 0\,,\ \text{otherwise}\end{cases}$, we get

$$P\left(W',\widehat{Q}\right)=\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{l=1}^{k} w_{jl}\|x_j-q_l\|^2+\|x_i-q_{l_0}\|^2\,,$$

that is
$P\left(W,\widehat{Q}\right)-P\left(W',\widehat{Q}\right)=\|x_i-q_{l_i}\|^2-\|x_i-q_{l_0}\|^2>0$
which contradicts the assumption that W minimizes $P\left(W,\widehat{Q}\right)$.

Note that in general, for any given $\widehat{Q}$ there are more solutions of $W^{(0)}$ type because any particular data x_i can be at minimum distance to more than one center of $\widehat{Q}$. ■

Theorem 2 *For any fixed $\widehat{W}$ satisfying the constrains of (10), the function $P\left(\widehat{W},Q\right)$ is minimized if and only if*

$$q_l=\frac{\sum\limits_{i=1}^{N}\widehat{w}_{il}x_i}{\sum\limits_{i=1}^{N}\widehat{w}_{il}},\quad i=\overline{1,k}.$$

Proof: For each $l=\overline{1,k}$ let $\mathcal{C}_l=\{x_i|x_i\in\mathcal{D},\widehat{w}_{il}=1\}$. Obviously $\mathcal{C}_1,\ldots,\mathcal{C}_k$ are pairwise disjoint clusters of $\mathcal{D}$ and $\sum\limits_{i=1}^{N}\widehat{w}_{il}=|\mathcal{C}_l|$, $\sum\limits_{i=1}^{N}\widehat{w}_{il}x_i=\sum\limits_{x_i\in\mathcal{C}_l}x_i\,,\ \ l=\overline{1,k}.$

Then

$$P\left(\widehat{W},Q\right)=\sum_{i=1}^{N}\sum_{l=1}^{k}\widehat{w}_{il}\|x_i-q_l\|^2=\sum_{l=1}^{k}\sum_{x_i\in\mathcal{C}_l}\|x_i-q_l\|^2.$$

Let $\sum\limits_{i=1}^{N}\widehat{w}_{il}x_i=\sum\limits_{\widehat{w}_{il}=1}x_i$ where

$q_l^{(0)}=\dfrac{1}{|\mathcal{C}_l|}\sum\limits_{x_i\in\mathcal{C}_l}x_i\,,\ \ l=\overline{1,k}.$

Obviously

$$\sum_{x_i\in\mathcal{C}_l}\|x_i-q_l\|^2=\sum_{x_i\in\mathcal{C}_l}\left\|x_i-q_l^{(0)}+q_l^{(0)}-q_l\right\|^2=\sum_{x_i\in\mathcal{C}_l}\left\|x_i-q_l^{(0)}\right\|^2+|\mathcal{C}_l|\left\|q_l^{(0)}-q_l\right\|^2+2\left(q_l^{(0)}-q_l\right)^T\left(\underbrace{\sum_{x_i\in\mathcal{C}_l}x_i-|\mathcal{C}_l|\,q_l^{(0)}}_{0}\right)\ge\sum_{x_i\in\mathcal{C}_l}\left\|x_i-q_l^{(0)}\right\|^2,$$

that is $P\left(\widehat{W},Q^{(0)}\right)\le P\left(\widehat{W},Q\right)$ for any $Q\in\mathcal{C}$. Moreover since the quality $P\left(\widehat{W},Q^{(0)}\right)=P\left(\widehat{W},Q\right)$ holds if and only if $q_l=q_l^{(0)}$ for all $l=\overline{1,k}$, for given $\widehat{W}$, $Q^{(0)}$ is the unique set of centers that minimizes $P\left(\widehat{W},Q\right)$. ■

The k-means algorithm viewed as an optimization process for solving (10) is as follows

The algorithm k-MOP

Input: $\mathcal{D}$ *- the data set,*
k *- the pre-specified number of clusters,*
d *- the data dimensionality,*
T *- threshold on the maximum number of iterations.*
Initializations: $Q^{(0)}$, $t\longleftarrow 0$
Solve $P\left(W,Q^{(0)}\right)$ *and get* $W^{(0)}$
$sw\longleftarrow false$
repeat
  $\widehat{W}\longleftarrow W^{(t)}$
  solve $P\left(\widehat{W},Q\right)$ *and get* $Q^{(t+1)}$
  if $P\left(\widehat{W},Q^{(t)}\right)=P\left(\widehat{W},Q^{(t+1)}\right)$ *then*
    $sw\longleftarrow true$
    output $\left(\widehat{W},Q^{(t+1)}\right)$
  else
    $\widehat{Q}\longleftarrow Q^{(t+1)}$
    solve $P\left(W^{(t)},\widehat{Q}\right)$ *and get* $W^{(t+1)}$

if $P\left(W^{(t)},\widehat{Q}\right)=P\left(W^{(t+1)},\widehat{Q}\right)$ *then*
$sw \longleftarrow true$
output $\left(W^{(t+1)},\widehat{Q},\right)$
endif
endif
$t \longleftarrow t+1$
until sw *or* $t>T$.

Note that the computational complexity of the algorithm k-**MOP** is $\mathcal{O}(Nkd)$ per iteration. The sequence of values $P\left(W^{(t)},\,Q^{(t)}\right)$ where $W^{(t)}$, $Q^{(t)}$ are computed by k-**MOP** is strictly decreasing, therefore the algorithm converges to a local minimum of the objective function.

4 The combined separating technique based on SVM and the k-means algorithm

At first sight, it seems unreasonable to compare a supervised technique to un unsupervised one mainly because they refer to totally different situations. On one hand the supervised techniques are applied in case the data set consists of correctly labeled objects, and on the other hand the unsupervised methods deal with unlabeled objects. However our point is to combine SVM and k-means algorithm, in order to obtain a new design of a linear classifier.

The aim of the experimental analysis is to evaluate the performance of the linear classifier resulted from the combination of the supervised SVM method and the 2-means algorithm.

Our method can be applied to whatever data, either linear separable or non-linear separable. Obviously in case of non-linear separable data the classification can not be performed without errors and in this case the number of misclassified examples is most reasonable criterion for performance evaluation. Of a particular importance is the case of linear separable data in this case the performance being evaluated in terms of both, misclassified examples and the generalization capacity expressed in terms of the width of separating area. In real live situations, usually is very difficult or even impossible to established whether the data represents a linear/non-linear separable set. In using the $SVM1$ approach we can identify which case the given data set belongs to. For linear separable data, $SVM1$ computes a separation hyperplane optimal from the point of view of the generalization capacity. In case of a non-linear separable data $SVM2$ computes a linear classifier that minimizes the number of misclassified examples. A series of developments are based on non-linear transform whose range is high dimensional space represented be kernel functions. The increase of dimensionality and the convenable choice of the kernel allow to transform the non-linear separable problem into a linear separable one. The computation complexity corresponding to kernel-based approaches is significantly large therefore in case the performance of the algorithm $SVM1$ proves reasonable good it could be taken as an alternative approach of a kernel-based SVM. We perform a comparative analysis on data consisting of examples generated from two dimensional Gaussian distributions.

In case of a non-linear separable data set using the k-means algorithm we get a system of pairwise disjoint clusters together with a set of their centers representing a local minimum point of the criterion (10), the clusters being linear separable when $k=2$. Consequently, the $SVM1$ algorithm computes a linear classifier that separates without errors the resulted clusters.

Our procedure is described as follows

Input: $\mathcal{S}=\left\{(x_i,y_i)|\, x_i\in\mathbf{R}^n, y_i\in\{-1,1\},\ i=\overline{1,N}\right\}$

Step 1. Compute the matrix $D=(d_{ik})$ *of entries*

$$d_{ik}=y_iy_k\left(x_i\right)^T x_k\,,\ i,k=\overline{1,N}\,,$$

and initialize $sh \longleftarrow true$

Step 2. If the constrained optimization problem

$$\begin{cases}\alpha^*=arg\left(\max\limits_{\alpha\in\mathbf{R}^N}\left(\alpha^T\mathbf{1}-\dfrac{1}{2}\,\alpha^TD\alpha\right)\right)\,,\\ \alpha_i\geq 0\,,\quad \forall\,1\leq i\leq N\,,\\ \sum\limits_{i=1}^{N}\alpha_iy_i=0\,,\end{cases}$$

do not have solution then
$sh \longleftarrow false$
input c, for hyperplane soft margin
Solve the constrained optimization problem

$$\begin{cases}\alpha^*=arg\left(\max\limits_{\alpha\in\mathbf{R}^N}\left(\alpha^T\mathbf{1}-\dfrac{1}{2}\,\alpha^TD\alpha-\dfrac{(\alpha_{max})^2}{4\,c}\right)\right)\,,\\ \alpha_i\geq 0\,,\quad \forall\,1\leq i\leq N\,,\\ \sum\limits_{i=1}^{N}\alpha_iy_i=0\,,\end{cases}$$

endif

Step 3. Select x_r, x_s *such that*

$$\alpha_r^*>0\,,\quad \alpha_s^*>0\,,\quad y_r=-1\,,\quad y_s=1$$

Compute the parameters w^*, b^* *of the separating hyperplane,*

$$\begin{cases} w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i \,, \\ b^* = -\frac{1}{2} (w^*)^T (x_r + x_s) \end{cases}$$

Compute the width of the separating area

$$\rho(w^*, b^*) = \frac{2}{\|w^*\|}$$

Step 4. if not sh then

compute nr_err1 *- the numbers of examples incorrect classified*

compute $err1$ *- error classification*

endif

Step 5. The set $\mathcal{D} = \left\{ x_i \mid x_i \in \mathbf{R}^d \,,\ i = \overline{1, N} \right\}$ *is divided in two clusters* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *using 2-means, marked out with* $y_i' = 1$ *and* $y_i' = -1$ *respectively.*

Step 6. Apply algorithm SVM1 for

$$\mathcal{S}' = \left\{ (x_i, y_i') \,\middle|\, x_i \in \mathbf{R}^d, y_i' \in \{-1, 1\}, i = \overline{1, N} \right\}$$

and obtain the parameters for optimal separating hyperplane: $w_1^*, b_1^*, \rho(w_1^*, b_1^*)$

compute nr_err2 *- the numbers of examples incorrect classified by* $2-means$

compute $err2$ *- error classification after* $2 - means$

Output: $w^*, b^*, \rho(w^*, b^*)$, nr_err1, $err1$, $w_1^*, b_1^*, \rho(w_1^*, b_1^*)$, nr_err2, $err2$.

5 Experimental results

In this section same of the results obtained in testing the potential of the methodology exposed in previous sections for solving classification problems. Some of the test were performed on simulated data randomly generated from two dimensional Gaussian distribution using kernels of polynomial and exponential types, respectively. A refined methodology is proposed in the final part of this section . The proposed methodology combines a Principal Component Analysis (PCA) approach for dimensionality reduction in the space of the initial data representations with kernel-based SVM to allow possible inseparability. The tests were performed on the free Wisconsin Diagnostic Breast Cancer Database, taken from UCI Machine Learning Repository: Data Sets, http://archive.ics.uci.edu/ml/datasets.html.

The result of the test performed on the famous XOR problem using the kernel $K(x, x') = (x^T x' + 1)^6$ is presented in Figure 1. The labeled data are

$$\mathcal{S} = \left\{ \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, -1 \right), \left(\begin{pmatrix} -1 \\ -1 \end{pmatrix}, -1 \right), \left(\begin{pmatrix} -1 \\ 1 \end{pmatrix}, 1 \right), \left(\begin{pmatrix} 1 \\ -1 \end{pmatrix}, 1 \right) \right\}$$

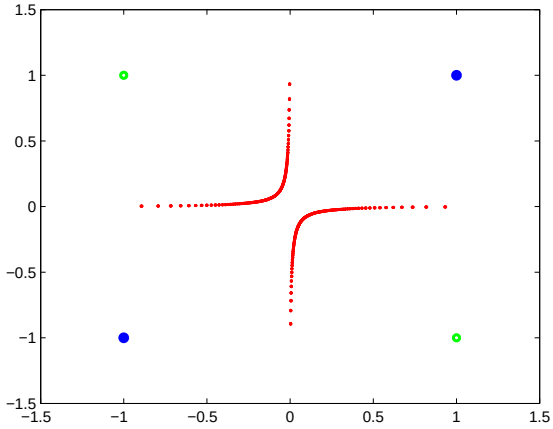

Figure 1: Solution of XOR computed by kernel SVM.

Several tests were performed in discriminated between examples coming from two classes when the data are generated from two dimensional Gaussian distribution. For instance, for samples of sizes $N_1 = 50$, $N_2 = 25$ respectively, generated from $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$ where

$$\mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mu_2 = \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$$

in most cases, we obtained non-linearly separable data. Some of the results obtained when we used different expression for polynomial an exponential kernels are represented in Figure 2 and Figure 3.

A long series of tests were performed on the medical data with confirmed diagnostic available at UCI Machine Learning Repository: Data Sets, http://archive.ics.uci.edu/ml/datasets.html. the dimension of each record is $d = 30$, for each example the confirmed diagnostic representing the presence/absence of breast cancer being also provided. The size of the data base is 569 examples, from which 357 are positive examples and 212 are negative examples and we used them for the SVM design and for testing purposes. The first series of tests were performed on design samples of sizes $20, 50, 100, 150, 200$ respectively, the example being taken randomly. In each case the complementary set of examples was used to test the performance of the computed discrimination function and the empirical error functions were computed. The results are presented in tables 1-8. For each sample a PCA analysis was developed separately for sub-samples coming from each class in order to determine their most informational directions. The tests pointed out that only the first 15 eigenvectors of the sample autocorrelation matrix are relevant, were we evaluated the relevance by the magnitude of the corresponding eigenvalues. We applied the SVM-based methodology using poly-

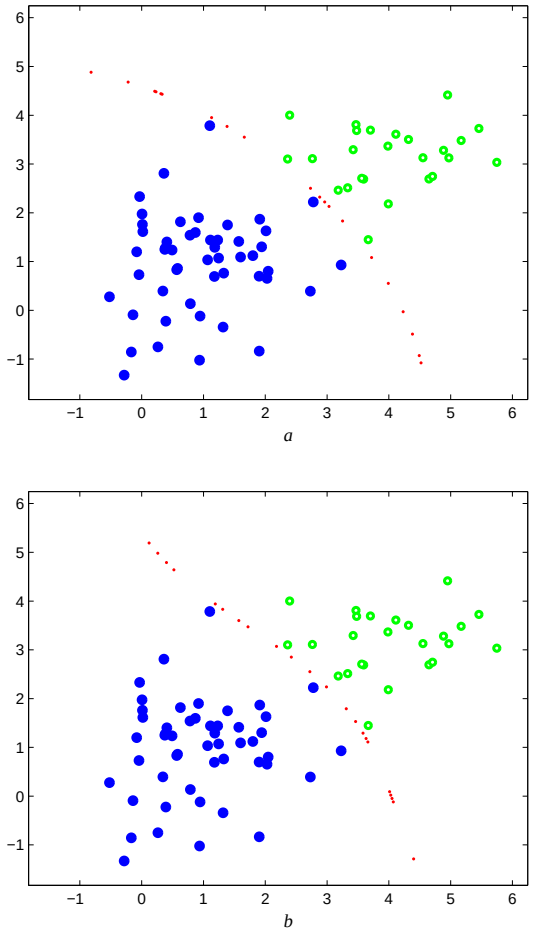

Figure 2: a) $K(x,x')=\left(x^Tx'+1\right)^2$; b) $K(x,x')=\left(x^Tx'+1\right)^6$.

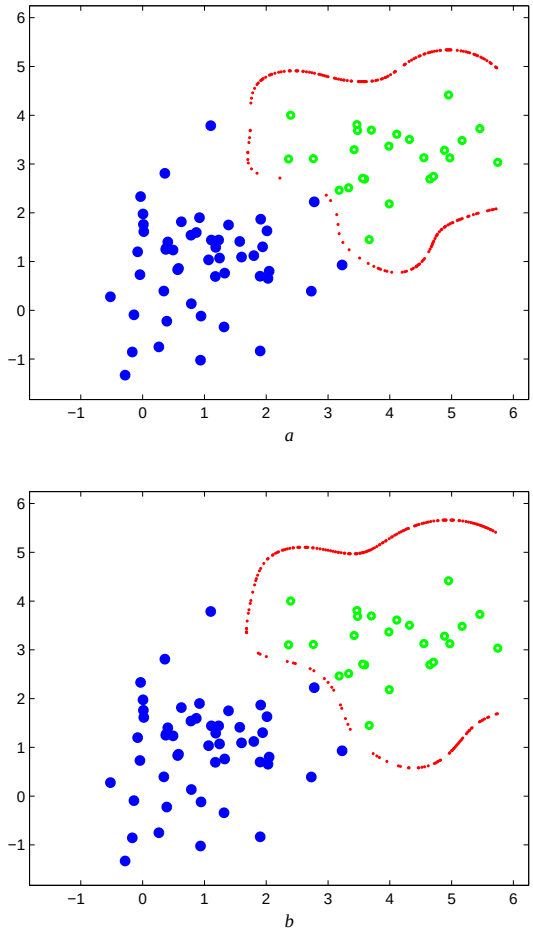

Figure 3: a) $K\left(x,x'\right)=\exp\left(-2\left(x-x'\right)^T\left(x-x'\right)\right)$; b) $K\left(x,x'\right)=\exp\left(-\left(x-x'\right)^T\left(x-x'\right)\right)$.

nomial kernels to the resulted 15 length representation obtained when we considered only the principal directions whose corresponding eigenvalues are larger than 10^{-3} and evaluated the empirical error functions. Our tests aimed to derived conclusions concerning the performance expressed in terms of the empirical error of the following classification schemes:

1. 2**-means classification scheme.**
   a. The 2-means algorithm applied to the whole data set.
   b. The 2-means minimum distance classification scheme (2-MMD). The data set is split into the design data set and the test data set. The 2-means algorithm is applied to the design data set, each element coming from the test data set is classified into the cluster whose center is at minimum distance.
2. **SVM classification scheme** using linear and exponential kernels (SVM_exp, SVM_lin). For each type of kernel the soft margin separating hyperplane is computed for the whole data set. The variant SVM_exp uses the exponential kernel $K\left(x,x'\right)=\exp\left(-\left(x-x'\right)^T\left(x-x'\right)\right)$ and the variant SVM_lin uses the linear kernel $K\left(x,x'\right)=\left(x-x'\right)^T\left(x-x'\right)$.
3. 2**-means–SVM classification schemes** ($SVM2m\_exp$, $SVM2m\_lin$). The 2-means algorithm is applied to the design data and the soft margin hyperplane is computed to separate the resulted clusters. The each of test example is then classified using the computed separating hyperplane. The variant $SVM2m\_exp$ uses the exponential kernel $K\left(x,x'\right)=\exp\left(-\left(x-x'\right)^T\left(x-x'\right)\right)$ and the variant $SVM2m\_lin$ uses the linear kernel

$$K\left(x, x'\right) = \left(x - x'\right)^T \left(x - x'\right).$$

The performance is evaluated by computing by the percentage of misclassified examples for design data set and test data set respectively. We used the following notation:

$E_{2m,p}$ − the percentage of misclassified examples when the 2-means algorithm is applied to the design data set;

$E_{2m,t}$ − the percentage of misclassified examples when the 2-means algorithm is applied to the test data set;

N_S − the number of support vector in case of a SVM-based classification schemes;

E_d − the percentage of misclassified examples when a SVM-based classification is applied to the design data set;

E_t − the percentage of misclassified examples when a SVM-based classification is applied to the design data set.

$CPUt$ − the duration expressed in seconds of computing the solution of the optimization problems (14) and (20) respectively using the SMO-algorithm. In table 1 the CPUt represents the duration expressed in seconds when the 2-means algorithm is applied.

Some of the results of our test s is presented in the initial data space of dimension 30 are presented in tables 1, 2, 3, 4.

Table 1: The 2-MMD scheme for $d = 30$.

N	$E_{2m,p}(\%)$	$E_{2m,t}(\%)$	$CPUt$
20	25	14.2	0.03
50	22	13.10	0.03
100	24	10.87	0.04

Table 2: The $SVM2m\_lin$ scheme for $d = 30$.

N	N_S	$E_d(\%)$	$E_t(\%)$	$CPUt$
20	2	25	15.3	1.24
50	3	22	11.94	27.2
100	3	24	10.44	317.86

Table 3: The $SVM2m\_exp$ scheme for $d = 30$.

N	N_S	$E_d(\%)$	$E_t(\%)$	$CPUt$
20	20	25	36.79	4.27
50	50	22	36.03	83.24
100	100	24	34.54	778.19

Table 4: The SVM_exp scheme for $d = 30$.

N	N_S	$E_d(\%)$	$E_t(\%)$	$CPUt$
20	20	0	36.79	4.35
50	50	0	36.03	86.58
100	100	0	34.55	777.11

Note that the $CPUt$ represents the total CPU time to apply the 2-means algorithm to the design data set and the classification of the test set examples.

The empirical error when the 2-means algorithm is applied to the whole data set is 14.58%. The tests entail several conclusion concerning the comparative analysis of the proposed classification schemes.

1. The best recognition can be obtained using the SVM_lin technique but unfortunately the time required to compute the soft margin hyperplane becomes prohibitively large when the volume of the design data set increases. In case the volume of the design data set is $N = 20$ the empirical error is 8.3% for the test data set and 0 for the design data set.

2. The 2-MMD and SVM2m_lin prove comparable performances from both points of view the encoding quality and the generalization capacities.

3. Also, in general, the SVM implementation using exponential kernels prove better performances the SVM_exp classification schemes proves poor generalization capacities.

4. The variant SVM2m_lin proves better generalization capacities as compared to the 2-MMD method but its duration increases significantly when volume of the design data becomes larger.

We combined the proposed methodology to a PCA preprocessing step. In case when we consider as being informative only the principal directions whose corresponding eigenvalues are larger than 10^{-3} we obtain that there are 15 principal directions for each class as well as for the whole data set, and in case the significance level is 10^{-2} we obtain that there are only

Table 5: The 2-MMD scheme for $d = 15$.

N	$E_{2m,p}(\%)$	$E_{2m,t}(\%)$	$CPUt$
20	25	14.2 0.03	
50	22	13.10	0.06
100	24	10.87	0.23

Table 6: The $SVM2m\_lin$ scheme for $d = 15$.

N	N_S	$E_d(\%)$	$E_t(\%)$	$CPUt$
20	2	25	15.3	1.26
50	3	22	11.94	27.76
100	3	24	10.44	310.55

Table 7: The $SVM2m\_exp$ scheme for $d = 15$.

N	N_S	$E_d(\%)$	$E_t(\%)$	$CPUt$
20	20	25	36.79	4.47
50	50	22	36.03	79.45
100	100	24	34.54	700.36

Table 8: The SVM_exp scheme for $d = 15$.

N	N_S	$E_d(\%)$	$E_t(\%)$	$CPUt$
20	20	0	36.79	4.19
50	50	0	36.03	85.03
100	100	0	34.55	773.15

12 principal directions respectively. We preprocessed data in order to obtain 15-length and 12-length representations and applied the described methodology. The PCA dimensionality reduction can by performed to ways. On one hand is to compute the principal directions for the whole data set, and on the other hand the principal directions can be computed separately for each class, in case of the second approach more informational representations result for all samples. The amount of time to compute the principal components of the whole data set is $t = 0.03$, the computation of the principal directions corresponding to each class being almost equal to 0.015. Being given that for this particular data set there are not significant differences between the representations resulted by applying these methods our option was to reduce the dimensionality by using the overall principal directions, and the results are presented in Table 5 - Table 8. Some other tests performed on different data bases pointed out that all the classification schemes proved better performance when the representation were computed in terms of the principal directions of each class.

Some of the obtained result are summarized in tables 5, 6, 7, 8. It is interesting to note that the empirical error of the 2-means algorithm applied in the reduced space remains unchanged that is the class separability is not affected by the dimensionality reduction.

Comparing the results presented in Table 1 - Table 4 and Table 5 - Table 8 we see that the performance of the classifier is significantly improved by including PCA as a preprocessing step. This can be interpreted as a proof that the effects of the minor components on the class variability is similar to some sort of noise that affects the data and is responsible for the decrease of classifier performances.

6 Summary and final remarks

The paper presents some results in using SVM-based techniques for classification purposes. Since the design of the SVM involves the solutions of quadratic programming problems a natural question is to find fast algorithms to solve the involved quadratic programming problems. In our developments we used the SMO algorithm introduced by Platt ([17]). We tried to include a PCA approach as a preprocessing step and apply the SVM-based methodology in the reduced space of principal components. The tests performed of UCI Machine Learning Repository: Data Sets, http://archive.ics.uci.edu/ml/datasets.html confirmed that this combined methodology allows significant improvement of the classification performance. Some work aiming to combine different types of classifiers with the SVM is still in progress and the results are going to be published elsewhere.

Acknowledgements: The research was supported was developed in the framework of the Ph.D. program al University of Pitesti, Romania.

References:

[1] S. Abe, *Support Vector Machines for Pattern Classification*, Springer-Verlag, 2005.

[2] C. Aviles, J. Villegas, J. Ocampo, R. Arechiga, Principal Components and Invariant Moments Analysis-Font Recognition Applied, *WSEAS Transactions on Computers*, Issue 5, Vol.5, 2006, pp. 1041–1046.

[3] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer Verlag, 2006.

[4] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery* 2, 1998, pp. 121-167.

[5] P.C. Cheng, B.C. Chien, W.P. Yang, Categorizing Medical Images by Supervised Machine Learning, *WSEAS Transactions on Computers*, Issue 12, Volume 5, 2006, pp. 3016-3021.

[6] C. Cortes and V.N. Vapnik, Support-vector networks, *Machine Learning* 20(3), 1995, pp. 273–297.

[7] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[8] G. Gan, C. Ma, J. Wu, *Data Clustering: Theory, Algorithms and Applications*, SIAM, 2007.

[9] N. El Gayar, S.A. Shaban, S. Hamdy, Face Recognition with Co-training and Ensemble-driven Learning, *WSEAS Transactions on Computers*, Issue 3, Vol. 6, 2007, pp. 507–513.

[10] S.R. Gunn, *Support Vector Machines for Classification and Regression*, University of Southampton, Technical Report, 1998.

[11] R. Herbrich, *Learning Kernel Classifiers: Theory and Algorithms*, MIT Press, 2002.

[12] G. Kaur, A.S. Arora,V.K. Jain, Multi-Class Support Vector Machine Classifier in EMG Diagnosis, *WSEAS Transactions on Signal Processing*, Issue 12, Volume 5, 2009, pp.379-389.

[13] J.B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley*, University of California Press, 281-297, 1967.

[14] J. Mercer, Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations, *Philosophical Transactions of the Royal Society of London* 209, Series A, 1909, pp. 415–446.

[15] H.Q. Minh, P. Niyogi and Y. Yao, Mercer's Theorem, Feature Maps and Smoothing, *Proc. of Computational Learning Theory (COLT'06)*, 2006, pp. 154–168.

[16] I. Paraschiv-Munteanu, Support Vector Machine in solving pattern recognition tasks, Technical Report, *Proceedings of First Doctoral Student Workshop in Computer Science*, University of Piteşti, May 2009.

[17] J.C. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, in *Advances Kernel Methods: Support Vector Machines*, edited by B. Schölkopf, C.J.C. Burges and A.J. Smola, The MIT Press, 1999, pp. 185–208.

[18] S.D. Sawarkar, A.A. Ghatol, Breast Cancer Malignancy Identification using Support Vector Machine, *WSEAS Transactions on Computers*, Issue 8, Volume 5, 2006, pp. 1707-1712.

[19] B. Schölkopf, A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.

[20] L. State and I. Paraschiv-Munteanu, A comparative analysis on the potential of SVM and k-means in solving classification tasks, *Proceedings of the First International Conference on Modelling and Development of Intelligent Systems MDIS-2009*, Sibiu, Romania, 2009, pp. 244–253.

[21] Z. Teng, F. Ren, S. Kuroiwa, The Emotion Recognition through classification with the Support Vector Machines, *WSEAS Transactions on Computers*, Issue 9, Volume 5, 2006, pp. 2008-2013.

[22] V.N. Vapnik, *Statistical Learning Theory*, New York, Wiley-Interscience, 1998.

[23] L. Wang, *Support Vector Machines: Theory and Applications*, Springer Verlang, 2005.

[24] H. Xaio, X. Zhang, Comparison Studies on Classification for Remote Sensing Image Based on Data Mining Method, *WSEAS Transactions on Computers*, Issue 5, Vol. 7, 2008, pp. 552–558.

[25] R. Xu, D.C.II Wunsch, *Clustering*, Wiley&Sons, 2009.

[26] C.Y. Yeh, S.J. Lee, C.H. Wu, S.H. Doong, A Hybrid Kernel Method for Clustering, *WSEAS Transactions on Computers*, Issue 10, Volume 5, 2006, pp. 2326-2333.