Professor Luminita STATE, PhD
Department of Mathematics and Informatics
University of Pitesti
E-mail: lstate@clicknet.ro
Professor Catalina COCIANU, PhD
E-mail: ccocianu@ase.ro
Marinela MIRCEA, PhD
Department of Informatics and Cybernetics
The Bucharest University of Economic Studies

IMPROVEMENTS OF THE RECOGNITION AND GENERALIZATION CAPACITIES OF THE NONLINEAR SOFT MARGIN SUPPORT VECTOR MACHINES

Abstract. *The aim of the paper is to develop a comparative analysis on the recognition and generalization capacities of some new variants of soft margin SVM resulted by refinements of weighting parameters and the bias. The core part of the paper is the fourth section, where a modified gradient ascent method for learning non-linear soft margin SVM is proposed. Several strategies for data driven control of the parameters in the learning of soft margin non-linear SVM are proposed in the fifth section and a comparative analysis of the recognition and generalization capacities of the resulted classifier against some of the most frequently used classifiers is developed. The experimental analysis was performed on artificially generated data as well as on Ripley and MONK's datasets reported in the fifth section of the paper. The tests confirmed substantial improvements from both point of views, recognition rate and generalization capacities as well as a faster convergence of the learning process. The final part contains a series of experimentally derived conclusions and some suggestions for further work.*

Keywords: *non-linear support vector machines, soft margin SVM, SMO Platt's algorithm, gradient-based learning, classifier design and evaluation, model-free learning, data driven control of parameters.*

JEL classification: C02, C14, C19, C45, C49, C61

1. Introduction

A classification problem can be modeled many ways, one of them can be described as follows. Assume that the aim is to discriminate among the instantiations of *m* concepts or classes, conventionally represented by the labels $h_1, \dots, h_m$. Each instantiation comes from one and only one concept, refered as the true provenance class. The usual representation of instantiations is in terms of a pre-selected finite

set of *n* descriptors or attributes. In the real world problems, one is forced to consider non-homogenous set of descriptors in the sense that some of them are nominals, other can be categorical or of continous type.
For simplicity sake, in the following we assume that the set of descriptors is homogenous, all of them being of continous type, therefore each instance can be represented by a particular point in $\mathbb{R}^n$ (the input space). The images of the concepts in $\mathbb{R}^n$ are conventionaly called classes. As an working assumption, the set of classes determined by a particular set of descriptors is taken as a partition of the input space. The structure of a classifier system involves three components, G (the generator), S (the system), and LM (the learning machine). The instances are generated by the component G using a certain sampling mechanism usually unknown to the observer. In a probabilistic framework, the unknown mechanism used by G to generate examples from $\mathbb{R}^n$ is modeled in terms of an unknown probability density function. The system S identifies for each example $x \in \mathbb{R}^n$ its provenance class $y(x) \in \{h_1, \dots, h_m\}$, the input/output dependency of S being unknown to the observer. The component LM implements a set Ω of hypotheses concerning the unknown input/output dependency of S and produces, for each example $x \in \mathbb{R}^n$ an output $\bar{y}(x, \omega) \in \{h_1, \dots, h_m\}$ representing a guess concerning $y(x)$ according to the current hypothesis $\omega \in \Omega$. Therefore each particular hypothesis ω can be viewed as a classifier of a certain type.

The set Ω is selected by the observer and can contain different types of classifiers, some of them being parametrized and some others parameter-free. The aim is to design a strategy (a learning algorithm) to find out the most suitable hypothesis in order to explain the behavior of S, corresponding to an inference about the unknown input-output dependency of the system, or equivalently the class structure in the input space, based on a certain finite sequence of observations $(x_i, y(x_i)), i = 1 \dots N$ taken on S. Consequently, a learning algorithm can be viewed as a search procedure in the space Ω, that is data driven in the sense that following each observation $(x_i, y(x_i))$, if ω is the current hypothesis then a new "more fitted" hypothesis is identified on the basis of $\{x_j, y(x_j), 1 \leq j \leq i\}$ and $\bar{y}(x_i, \omega)$, where the performance is expressed in terms of a certain criterion function. Moreover, the found hypothesis should assure correct discrimination among the classes in the input space, that is to provide good generalization capacities.
The criterion function evaluates, at each moment, the quality of the current hypothesis in approximating the unknown input/output dependency of S on the

basis of the evidence represented by the pairs $(x_i, y(x_i))$ still observed. At the end of the searching process, the quality of the learned hypothesis ω_0 in approximating the input/output functionality of S depends on the particular set of hypotheses Ω as well as on the sequence $(x_1, \ldots, x_N)$ generated by G. Although in principle Ω can be assumed to be an universal set, in real world applications the set of hypotheses is such that to contain only "simple" elements in the sense that each element is a simple parametrized functional dependency on the entries of the input, usually of linear type, and the learning process is an adaptive way to adjust the parameters while new observations are collected. The most frequently case is the binary classification, when the examples come from two classes, the output of S being either 1 or -1.

Since 1957, when the first adaptive learning procedure, known as the perceptron algorithm, was proposed by Rosenblatt (Rosenblatt, 1957), there have been proposed a long series of more refined learning methods, most of them of gradient type (Levenberg, 1944; Marquardt, 1963) (Ho, Kashyap 1965; Ho, Kashyap 1966). Some alternative methods exploit the whole bunch of observations in order to identify a separating surfaces among the samples coming from different classes. In other words, the best hypothesis for being implemented at the level of the LM component is computed off-line using the global information contained by the yet-seen samples.

For simplicity sake, the functional expression of the desired separating surfaces is of linear type, that is the design of the classifier is such that the examples coming from different classes are separated by hyperplanes. Unfortunately, very often the subclasses of examples coming from different classes cannot be separated by hyperplanes and moreover the cost of designing more refined classifiers of non-linear type is too high. Recently, the methods based on kernels supplied reasonable solutions in these cases (Abe, 2010; Shawe-Taylor, Cristianini, 2000; Shawe-Taylor, Cristianini, 2004;. Liu, Principe, Haykin, 2010).

Of a crucial importance are the contributions of Vapnik (Vapnik, 1995; Vapnik, 1998), founding the statistical learning theory. One of the most efficient classifier from the point of view of generalization capacities is Support Vector Machine (SVM), also introduced by Vapnik.

The aim of the paper is to develop a comparative analysis on the recognition and generalization capacities of some new variants of soft margin SVM resulted by refinements of weighting parameters and the bias. The introductory part is followed by a brief presentation of the non-linear SVM and soft margin SVM supplied in the first two sections. The core part of the paper is represented by the fourth section, where a modified gradient ascent method for learning non-linear soft margin SVM is presented. In order to solve the resulted QP-problem, a variant of Platt's SMO algorithm is presented in the final part of the fourth section. Several strategies for data driven control of the parameters in the learning of soft margin non-linear SVM

are proposed in the fifth section together with a comparative analysis of the recognition and generalization capacities of the resulted classifier against some of the most frequently used classifiers. The tests were performed on artificially generated data as well as on two standard datasets, Ripley and MONK's. The tests pointed out that the variation of the recognition rates depends also on the inner structure of the classes from which the learning data come as well as on their separability degree. The final part of the paper contains a series of experimentally derived conclusions and some suggestions for further work.

2. The non-linear SVM

Let us denote by $\mathcal{S} = \{(x_i, y_i),\ x_i \in \mathbb{R}^n, y_i \in \{-1,1\}, 1 \leq i \leq N\}$, the collection of examples coming from two classes, each pair (x_i, y_i) representing an example x_i generated by the G component and y_i the output of S, the true provenance class. We say that $\mathcal{S}$ is linearly separable if there exists a hyperplane in the space of inputs separating the subset of examples for which S emits (-1) of the subset for which S emits (1). In real world applications, it is either very difficult or even impossible to check whether $\mathcal{S}$ is linearly separable.

On one hand, the information concerning the classes in the input space is provided exclusively by the examples generated by G and fed to S. Consequently, even when $\mathcal{S}$ happens to be linearly separable, there are no reasons to assume that the classes in the input space are also linearly separable. On the other hand, we would like to implement at the level of the LM component a classifier having good generalization capacities, that is to discriminate as well as possible not only between the known examples provided by G, but for new unseen examples coming from the same classes.

In order to extract as much information as possible from $\mathcal{S}$ concerning the incompletely known classes in the input space, possibly residing from the hidden structure of the set of selected descriptors, a non-linear transform of the given data onto a new space is hoped to be useful in order to reveal some new information concerning these classes. A second reason to look for a non-linear transform projecting the collection $\mathcal{S}$ onto a new space comes from the fact that the separability degree between images of the classes in the new space could be increased. In such a case, obviously, the separability degree between the subsets of examples from $\mathcal{S}$ for which S computes 1, -1 respectively is also increased, sometimes the images of these subsets becoming linearly separable.

From mathematical point of view, the non-linear transform is a vector valued function $g: \mathbb{R}^n \to \mathcal{F}$, the image of $\mathcal{S}$ in the space $\mathcal{F}$ being given by the set of new representations of the given data

$\mathcal{S}_g = \{(g(x_i), y_i),\ x_i \in \mathbb{R}^n, y_i \in \{-1,1\}, 1 \leq i \leq N\}$. The transform g is referred as a feature extractor, and $\mathcal{F}$ is called the feature space, its dimension being not necessarily finite.

Assuming that the $\mathcal{S}_g$ is at least "almost linearly separable", it appears quite natural to use linear classifiers in the feature space, that is to implement at the level of the LM component a set of hypothesis Ω, where each particular hypothesis $\omega \in \Omega$ corresponds to the parameters of a linear classifier in the space $\mathcal{F}$. When the dimension of $\mathcal{F}$ is finite, say m, a hypothesis ω corresponds to a pair $(w, b) \in \mathbb{R}^m \times \mathbb{R}$, and, for the input x, according to ω, the LM component computes the output $\bar{y}(x, \omega) = \begin{cases} 1, w^T g(x) + b \geq 0 \\ -1, w^T g(x) + b < 0 \end{cases}$.

The performance of the resulted classifier is essentially determined by the feature extractor g, as well as by the particular parameter(w, b). Concerning the design of the feature extractor g, the main problem is to select a particular functional expression of g, such that, on one hand, $\mathcal{S}_g$ is almost linearly separable, and on the other hand the computational complexity involved by the estimation process of the parameter (w, b) is kept at a reasonable level. The "kernel trick" provides a solution to these problems. It consists in selecting a function K that "covers" the explicit functional expression of g, therefore the evaluation of the expression $(w^T g(x) + b)$ is performed exclusively in terms of K. Since g is "hidden" by K, the resulted feature space cannot be explicitly known, therefore its dimension may be even infinite. The core result in approaches of this type is the celebrated theorem due to Mercer (Mercer, 1908). According to this results, if $K: \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty)$ is a continuous symmetric function, the existence of a function g such that for any $x, x' \in \mathbb{R}^n, K(x, x') = g(x)^T g(x')$ holds, is guaranteed in case K satisfies a set of quite general conditions. A series of particular expressions of kernels satisfying the Mercer's conditions have been extensively used in the published literature (Abe, 2010). In the following we use two of them, namely the Gauss Radial Basis Function (GRBF), $K(x, x') = exp(-\gamma \|x - x'\|^2), \gamma > 0$ and the Exponential Radial Basis Function (ERBF),

$K(x, x') = exp(-\gamma \|x - x'\|), \gamma > 0$.

Since $\mathcal{S}_g$ is finite, in case it is linearly separable in the space $\mathcal{F}$, there are an infinite number of classifiers ω that separate the given data without errors.

Let us assume that for a selected kernel K, $\mathcal{S}_g$ is linearly separable. Then we could

search for a linear classifier in $\mathcal{F}$ that offers the best generalization capacity in the sense that it still classifies at least "almost correctly", new, unseen yet examples. This requirement may be formulated as the task to determine the parameters (w, b) such that the hyperplane of equation $w^T g(x) + b = 0$ is as equidistant as possible to all images of the training data in the feature space, therefore it is aimed to separate the examples of $\mathcal{S}_{\mathcal{F}}$ with the largest "gap" between positive and negative examples. Such a classifier is referred as an optimal margin classifier.

Stated in mathematical terms, the problem is formulated as follows. Let *K* be a kernel and *g* be the induced feature extractor, An optimal margin classifier is a solution of the constrained QP problem (Vapnik, 1995),

$$\begin{cases} minimize\ \frac{1}{2}\|w\|^2 \\ y_i(w^T g(x_i) + b) \geq 1, 1 \leq i \leq N \end{cases} \quad (1)$$

its corresponding dual problem being the constrained QP problem imposed on the objective function $Q(\alpha)$,

$$\begin{cases} maximize\ Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j\, y_i y_j K(x_i, x_j) \\ \sum_{i=1}^{N} \alpha_i\, y_i = 0 \\ \alpha_i \geq 0, 1 \leq i \leq N \end{cases} \quad (2)$$

Let $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_N^*)^T$ be a solution of (2). In case $\alpha_i^* > 0$, the example x_i is called a support vector (Vapnik, 1995). Since the solutions of (2) do not involve the parameter *b*, its value should be determined such that $1 - \min_{i, y_i=1} w^{*T} g(x_i) \leq b^* \leq -1 - \max_{i, y_i=-1} w^{*T} g(x_i)$ holds, therefore more options concerning b^* are allowed. If $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_N^*)^T$ is a solution of (2), then the optimal hypothesis ω^* corresponds to (w^*, b^*), where

$$w^* = \sum_{i=1}^{N} \alpha_i^*\, y_i g(x_i), \quad (3)$$

and one of the most used expression of b^* is (Abe, 2010; Shawe-Taylor, Cristianini, 2000)

$$b^* = -\frac{1}{2}\left\{ \max_{i, y_i=-1} \sum_{j=1}^{N} \alpha_j^*\, y_j K(x_j, x_i) + \min_{i, y_i=1} \sum_{j=1}^{N} \alpha_j^*\, y_j K(x_j, x_i) \right\} \quad (4)$$

Being given the sequence of observations $\mathcal{S}$, according to the optimal hypothesis ω^*, in case a new input x is provided by G, the computed output of the LM component is $\bar{y}(x,\omega^*) \in \{-1,1\}$, where $\bar{y}(x,\omega^*) = 1$ if and only if $\sum_{j=1}^{N} \alpha_j^* y_j K(x_j, x) + b^* \geq 0$.

3. Soft Margin SVM

Let us assume now that for a selected kernel K, $\mathcal{S}_\Phi$ is non-linearly separable. Then we could search for a classifier (w, b) such that according to its corresponding hypothesis ω, the LM component "imitates" as much as possible the behavior of S. This idea can be formulated in mathematical terms as follows.

Let us denote by g the feature extractor such that $K(x,x') = g(x)^T g(x')$ and ω a hypothesis corresponding to the linear classifier in the feature space of parameter (w, b), that is, for the input x, the output of the LM component is $\bar{y}(x,\omega) = 1$ if and only if $w^T g(x) + b \geq 0$. The model of the non-linear SVM can be extended by including the slack variables $\xi_1, \xi_2, \ldots, \xi_N$, where ξ_i expresses the magnitude of the error committed by ω for the observation (x_i, y_i), that is $\xi_i = max\{0, 1 - y_i(w^T g(x_i) + b)\}$.

For any misclassified example (x_i, y_i), the value of ξ_i expresses the magnitude of the error committed by the hypothesis ω with respect to (x_i, y_i). The overall importance of the cumulated errors usually can be expressed as

$$F\left(\sum_{i=1}^{N} \xi_i^t\right) \tag{5}$$

where F is a convex and monotone increasing function and $t > 0$ is a weight parameter.

Therefore, by combining additively the objective function of the problem (1) with the overall effect of the errors (5), we obtain a new QP problem (Cortez, Vapnik, 1995)

$$\begin{cases} minimize \left\{\frac{1}{2}\|w\|^2 + CF\left(\sum_{i=1}^{N} \xi_i^t\right)\right\} \\ y_i(w^T g(x_i) + b) \geq 1 - \xi_i, 1 \leq i \leq N \\ \xi_i \geq 0, 1 \leq i \leq N \end{cases} \tag{6}$$

where C is a conventionally selected constant used to weight the effect of the cumulated errors.
Being given its complexity, the problem (6) cannot be solved in this general form, but only for particular functional expressions of F and the weight parameter t. The simplest model uses $F(u) = u$ and $t = 1$, the problem (6) becoming the constrained QP-problem

$$\begin{cases} minimize \left\{ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \right\} \\ y_i(w^T g(x_i) + b) \geq 1 - \xi_i, 1 \leq i \leq N \\ \xi_i \geq 0, 1 \leq i \leq N \end{cases} \quad (7)$$

whose dual QP-problem is

$$\begin{cases} maximize\ Q(\propto) \\ \sum_{i=1}^{N} \propto_i y_i = 0 \\ 0 \leq \propto_i \leq C, \qquad 1 \leq i \leq N \end{cases} \quad (8)$$

where

$$Q(\propto) = \sum_{i=1}^{N} \propto_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \propto_i \propto_j y_i y_j K(x_i, x_j)$$

As in case of the non-linear SVM, if $\propto^* = (\propto_1^*, \propto_2^*, ..., \propto_N^*)^T$ is a solution of (8), then the parameters (w^*, b^*) are given by (3) and (4).

4. Learning Algorithms of Nonlinear Soft Margin SVM

According to the arguments supplied in the Section 3, the computation of the soft margin SVM separating hyperplane corresponds to solving the QP-problem (8). In this section we present a modified gradient ascent method and a variant of the Platt's SMO algorithm to approximate a solution of (8).

4.1. Modified Gradient Ascent Method for Learning Nonlinear Soft Margin SVM

The learning rule of gradient ascent type for linear SVM proposed in (State, Cocianu, Vlamos, 2011) can be extended to the non-linear case of Soft Margin SVM as follows. Let $\mathcal{S} = \{(x_i, y_i),\ x_i \in \mathbb{R}^n, y_i \in \{-1,1\}, 1 \leq i \leq N\}$ be the set of observations taken on S. For simplicity sake, we assume that the outputs of the first *m*, *N-m* inputs are 1, -1 respectively. By straightforward computations, the entries of the gradient $\nabla_\alpha Q(\alpha)$ and the Hessian matrix $H(Q(\alpha)) = \left\| \frac{\partial^2 Q(\alpha)}{\partial \alpha_k \partial \alpha_p} \right\|$ are

$$\frac{\partial Q(\propto)}{\partial \propto_k} = 1 - y_k \sum_{i=1}^{N} \propto_i y_i K(x_k, x_i),\ 1 \leq i \leq N \quad (9)$$

$$\frac{\partial^2 Q(\alpha)}{\partial \alpha_k \partial \alpha_p} = -y_p y_k K(x_k, x_p) \tag{10}$$

Note that $H(Q(\alpha))$ is a negative semi-defined matrix. If α^{old} is the current value of the parameter α, then the updating rule of a gradient type learning algorithm is

$$\alpha = \alpha^{old} + \rho \nabla_\alpha Q(\alpha)|_{\alpha=\alpha^{old}} \tag{11}$$

where $\rho > 0$ is the learning rate.

Since the parameter $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$ has to satisfy the constraints $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $0 \le \alpha_i \le C,\ 1 \le i \le N$, the updating rule (11) should be modified to assure that the updated parameter still belongs to the space of the feasible solutions of (8). Consequently, we propose a modified learning rule which is a tradeoff of the gradient ascent method and the requirements imposed by the constraints of (8). In our approach, we select two entries to be modified; let p_1, p_2 be the indices of the entries in the current parameter vector α^{old} selected for being updated, $1 \le p_1 \le m,\ m+1 \le p_2 \le N$. We denote by $\rho_1 \in [0,1]$ a weighting parameter expressing the relative "influence" of $\frac{\partial Q(\alpha)}{\partial \alpha_{p_1}}\Big|_{\alpha=\alpha^{old}}$ and $\frac{\partial Q(\alpha)}{\partial \alpha_{p_2}}\Big|_{\alpha=\alpha^{old}}$ on the direction of the updating displacement. Consequently, according to the gradient ascent rule, the updated values of the entries $\alpha_{p_i}, i = 1,2$ should be given by (11). However, one or both updated values could fail to satisfy the constraints of the QP-problem (8). Consequently, the updating rule should assure that the new parameter α still satisfies the constraints and the search direction is selected such that to maximize $Q(\alpha) - Q(\alpha^{old})$. Let us assume that the search direction represented by (p_1, p_2) is somehow determined. Then the entries of the updated parameter α are

$$\begin{cases} \alpha_{p_i} = \alpha_{p_i}^{old} + d, & i = 1,2 \\ \alpha_p = \alpha_p^{old}, & p \neq p_1, p \neq p_2 \end{cases} \tag{12}$$

where

$$\Delta = \rho\left(\rho_1 \frac{\partial Q(\alpha)}{\partial \alpha_{p_1}}\Big|_{\alpha=\alpha^{old}} + (1-\rho_1)\frac{\partial Q(\alpha)}{\partial \alpha_{p_2}}\Big|_{\alpha=\alpha^{old}}\right)$$

$$d_i = \begin{cases} \Delta, \alpha_{p_i}^{old} + \Delta \le C \\ C - \alpha_{p_i}^{old}, otherwise \end{cases}, i = 1,2 \text{ and}$$

$d = \min\{d_1, d_2\}$. Note that the search direction (p_1, p_2) should satisfy the addi-

tional constraints $\left.\frac{\partial Q(\alpha)}{\partial \alpha_{p_1}}\right|_{\alpha=\alpha^{old}} \geq 0$, $\left.\frac{\partial Q(\alpha)}{\partial \alpha_{p_2}}\right|_{\alpha=\alpha^{old}} \geq 0$, and $\Delta > 0$. The fact that the updated parameter satisfies the constraints of the problem (8) stems from the following arguments. First of all, since $\sum_{i=1}^{N} \alpha_i^{old} y_i = 0$ and $y_{p_1} * y_{p_2} = -1$, according to (12) we get $\sum_{i=1}^{N} \alpha_i y_i = 0$. Also, $\Delta > 0$ implies $0 \leq \alpha_i,\ 1 \leq i \leq N$. Now we have to argue that the conditions $\alpha_i \leq C,\ 1 \leq i \leq N$ also hold. Indeed, we distinguish the following cases

a) If $\alpha_{p_i}^{old} + \Delta \leq C,\ i = 1,2$, then $d = \Delta$, therefore $\alpha_i \leq C,\ 1 \leq i \leq N$

b) If $\alpha_{p_1}^{old} + \Delta > C$ and $\alpha_{p_2}^{old} + \Delta \leq C$, then $d_1 = C - \alpha_{p_1}^{old} \geq 0$, $d_2 = \Delta$, therefore $0 \leq d_1 = C - \alpha_{p_1}^{old} < \Delta = d_2$, hence $d = d_1$. Consequently, $\alpha_{p_1} = C$ and $\alpha_{p_2} = \alpha_{p_2}^{old} + C - \alpha_{p_1}^{old} \leq \alpha_{p_2}^{old} + \Delta \leq C$. Obviously, a similar argument holds in case $\alpha_{p_2}^{old} + \Delta > C$ and $\alpha_{p_1}^{old} + \Delta \leq C$

c) If $\alpha_{p_1}^{old} + \Delta > C$ and $\alpha_{p_2}^{old} + \Delta > C$, then $d_1 = C - \alpha_{p_1}^{old} \geq 0$, $d_2 = C - \alpha_{p_2}^{old} \geq 0$, that is $\alpha_{p_i} = \alpha_{p_i}^{old} + min\{C - \alpha_{p_1}^{old}, C - \alpha_{p_2}^{old}\} \leq \alpha_{p_i}^{old} + C - \alpha_{p_i}^{old} = C, i = 1,2$

The indices p_1, p_2 involved in the updating step should be selected such that to assure the local maximization of $\left(Q(\alpha) - Q\left(\alpha^{old}\right)\right)$. Using first order approximations,

$$Q(\alpha) \cong Q\left(\alpha^{old}\right) + \left(\alpha - \alpha^{old}\right)^T \left(\nabla Q(\beta)|_{\beta=\alpha^{old}}\right) =$$

$$= Q\left(\alpha^{old}\right) + \left(\alpha_{p_1} - \alpha_{p_1}^{old}\right)\left(\left.\frac{\partial Q(\beta)}{\partial \beta_{p_1}}\right|_{\beta=\alpha^{old}}\right) + \left(\alpha_{p_2} - \alpha_{p_2}^{old}\right)\left(\left.\frac{\partial Q(\beta)}{\partial \beta_{p_2}}\right|_{\beta=\alpha^{old}}\right) =$$

$$= Q\left(\alpha^{old}\right) + d\left[\left(\left.\frac{\partial Q(\beta)}{\partial \beta_{p_1}}\right|_{\beta=\alpha^{old}}\right) + \left(\left.\frac{\partial Q(\beta)}{\partial \beta_{p_2}}\right|_{\beta=\alpha^{old}}\right)\right]$$

Consequently, in order to assure increasing values of the objective function, the pair of indices (p_1, p_2) should be such that the following conditions hold (State, Cocianu, Vlamos, 2011)

$$\begin{cases} 1 \leq p_1 \leq m, \quad m+1 \leq p_2 \leq N \\ \Delta > 0 \\ \left[\left(\left.\frac{\partial Q(\beta)}{\partial \beta_{p_1}}\right|_{\beta=\alpha^{old}}\right) + \left(\left.\frac{\partial Q(\beta)}{\partial \beta_{p_2}}\right|_{\beta=\alpha^{old}}\right)\right] > 0 \end{cases} \tag{13}$$

The implementation of the resulted adaptive learning algorithm uses a given parameter $\varepsilon > 0$ controlling the accuracy such that when at least one of the conditions (14), (15) holds, the learning process is stopped.

$$\Delta\left[\left(\frac{\partial Q(\beta)}{\partial \beta_{p_1}}\Big|_{\beta=\alpha^{old}}\right)+\left(\frac{\partial Q(\beta)}{\partial \beta_{p_2}}\Big|_{\beta=\alpha^{old}}\right)\right] \leq 0 \qquad (14)$$

$$\text{for all } 1 \leq p_1 \leq m,\ m+1 \leq p_2 \leq N$$

$$\left|Q(\alpha) - Q\left(\alpha^{old}\right)\right| < \varepsilon \qquad (15)$$

4.2. A Variant of Platt's SMO algorithm

The aim is to propose a variant of the Platt's SMO algorithm for solving the dual QP-problem (8).

Sequential minimal optimization (SMO) algorithm was introduced by Platt (J. Platt, 1998) and further extended by several authors (Keerthi, Shevade, 2003; Knebel, Hochreiter, Obermayer, 2008) is a simple algorithm that allows to solve the SVM-QP problem without extra-matrix storage by decomposing the overall QP-problem into simple QP sub-problems similar to Osuna's method (Osuna, Freund, Girosi, 1997). The idea of the SMO algorithm (J. Platt, 1998) is to solve the smallest optimization problem at each step, in case of the QP-problem corresponding to the soft margin SVM, the smallest optimization sub-problem involving only two Lagrange multipliers. The reason of optimizing two Lagrange multipliers comes from the requirement that the entries of the parameter α should satisfy the constraint $\sum_{i=1}^{N} \alpha_i\, y_i = 0$. At every step, according to the SMO algorithm, two Lagrange multipliers are selected to jointly optimize, the optimal values for these multipliers are found and performs updates to reflect the new optimal values. The computation can be briefly described as follows. Being selected two Lagrange multipliers, SMO computes the constraints on these multipliers and solves for the constraint maximum. The bound constraints cause the Lagrange multipliers to lie within a box, while the linear quality constraint causes the Lagrange multipliers to lie on a diagonal line, that is the constraint maximum of the objective function must lie on a diagonal line segment too.

Let K be a Mercer kernel and g its corresponding feature extractor, that is $K(x,x') = g(x)^T g(x'), \forall x, x' \in \mathbb{R}^n$. Let us denote by $f_\alpha(x) = w^T g(x) + b$, where $w = \sum_{i=1}^{N} \alpha_i\ y_i g(x_i)$ is the parameter of a separating hyperplane. Then

$$f_\alpha(x) = b + \sum_{i}^{N} y_i\, \alpha_i\, K(x, x_i) \qquad (16)$$

The idea of the SMO algorithm is to use a predefined constant $C > 0$, and

a tolerance parameter $\tau > 0$, expressing a sort of tradeoff between accuracy and efficiency. At each step two examples (x_p,y_p), (x_q,y_q) are looked for such that the following condition holds,

$$\begin{aligned}&(f_\alpha(x_p)-y_p+\tau < f_\alpha(x_q)-y_q-\tau)\\ &\wedge\big((\alpha_p < C \wedge y_p = 1)\vee(\alpha_p > 0 \wedge y_p = -1)\big)\\ &\wedge\big((\alpha_q < C \wedge y_q = -1)\vee(\alpha_q > 0 \wedge y_q = 1)\big)\end{aligned} \quad (17)$$

Let us assume that, at the current step, there exists at least a pair (x_p,y_p), (x_q,y_q) for which (17) holds. The entries $\propto_p$ and $\propto_q$ of the current parameter α are modified such that to increase $f_\alpha(x_p)$and to decrease$f_\alpha(x_q)$.

Since the updated parameter has to fulfill the constraint $\sum_{i=1}^{N}\alpha_i y_i = 0$, the updating rules are,

$\propto_p \leftarrow \propto_p + y_p\eta$

$\propto_q \leftarrow \propto_q - y_q\eta$

where

$$\eta = \frac{(f_\propto(x_q)-y_q)(f_\propto(x_p)-y_p)}{K(x_p,x_p)-2K(x_p,x_q)+K(x_q,x_q)} \quad (18)$$

If the conditions $0 \leq \propto_p + y_p\eta \leq C$ and $0 \leq \propto_q - y_q\eta \leq C$ do not hold, the value of the tolerance parameter η should be decrease accordingly. In case, at a certain step, there are no examples (x_p,y_p), (x_q,y_q) such that (17) holds, the search process is stopped.

Our variant of the Platt's SMO algorithm uses the following updating rules. Let (x_p,y_p), (x_q,y_q) be a pair of examples such that (17) holds, and $\propto^{old}$the current parameter. Then,

$\propto_p = \propto_p^{old} + y_p\eta$

$\propto_q = \propto_q^{old} - y_q\eta$

where η is given by (18). The value of the parameter η should be adjusted to assure that the updated values $\propto_p$ and $\propto_q$ still belong to $[0, C]$. Our option is for the following adjusting strategy. Assume that at least one of the entries $\propto_p,\propto_q$ does not belong to $[0, C]$.

a) If $y_p = y_q = 1$, then η is set to $min\left(\alpha_q^{old}, C - \alpha_p^{old}\right)$. Indeed, since $\eta > 0$, $\alpha_p > 0$ and $\alpha_q < C$, therefore at least one of the entries α_p, α_q does not belong to $[0, C]$ means that at least one of the inequalities $\alpha_p > C$, $\alpha_q < 0$ holds.

a1.If $\alpha_p^{old} + \alpha_q^{old} < C$, then setting $\eta = \alpha_q^{old}$, we get $0 \leq \alpha_p = \alpha_p^{old} + \alpha_q^{old} < C$, and $\alpha_q = 0$.

a2. If $\alpha_p^{old} + \alpha_q^{old} \geq C$, then setting $\eta = C - \alpha_p^{old}$, we get $\alpha_p = C$, and $\alpha_q = \alpha_q^{old} - \left(C - \alpha_p^{old}\right) \geq 0$.

Obviously $\alpha_q^{old} - \left(C - \alpha_p^{old}\right) = \alpha_q^{old} + \alpha_p^{old} - C \leq 2C - C = C$

b) If $y_p = 1, y_q = -1$ then η is set to $C - max\left\{\alpha_p^{old}, \alpha_q^{old}\right\}$. In order to prove that the new setting of η assures that α_p and α_q belong to $[0, C]$, let us analyze the following two cases. Obviously, in this case at least one of the entries α_p, α_q does not belong to $[0, C]$, that is $\alpha_p > C$ and/or $\alpha_q > C$.

b1. If $\alpha_p^{old} < \alpha_q^{old}$, then taking $\eta = C - \alpha_q^{old}$ we get $\alpha_p = \alpha_p^{old} + C - \alpha_q^{old} = C - \left(\alpha_q^{old} - \alpha_p^{old}\right) < C$, $\alpha_q = C$, and $\alpha_p \geq \alpha_p^{old} \geq 0$.

b2. If $\alpha_p^{old} \geq \alpha_q^{old}$ then setting $\eta = C - \alpha_p^{old}$ and using similar arguments we get that both updated entries belong to $[0, C]$.

c) If $y_p = -1, y_q = 1$ then η is set to $min\left\{\alpha_p^{old}, \alpha_q^{old}\right\}$. In this case when the condition that both α_p, α_q belong to $[0, C]$ is violated, then at least one of the inequalities $\alpha_p < 0, \alpha_q < 0$ holds.

c1. If $\alpha_p^{old} > \alpha_q^{old}$, then $\eta = \alpha_q^{old}$, therefore $0 < \alpha_p = \alpha_p^{old} - \alpha_q^{old} \leq C$ and $\alpha_q = 0$.

c2. Similarly, if $\alpha_p^{old} \leq \alpha_q^{old}$ then $\eta = \alpha_p^{old}$ so $0 \leq \alpha_q = \alpha_q^{old} - \alpha_p^{old} \leq C$ and $\alpha_p = 0$.

d) If $y_p = y_q = -1$, then η is set to $min\left(\alpha_p^{old}, C - \alpha_q^{old}\right)$. In this case at least one of the inequalities $\alpha_q > C$, $\alpha_p < 0$ holds.

d1.If $C - \alpha_q^{old} < \alpha_p^{old}$, then $\eta = C - \alpha_q^{old}$, and using the obvious relation $C < \alpha_p^{old} + \alpha_q^{old} \leq 2C$ we get $0 < \alpha_p = \alpha_p^{old} + \alpha_q^{old} - C \leq C$, and $\alpha_q = C$.

d2. If $C - \alpha_q^{old} \geq \alpha_p^{old}$, then $\eta = \alpha_p^{old}$, therefore $\alpha_p = 0$, and $0 \leq \alpha_q = \alpha_q^{old} + \alpha_p^{old} \leq C$.

The implementation of this variant of the Platt's SMO algorithm uses the stopping condition $\mathcal{C}$ defined in terms of the tolerance parameter $\tau > 0$ and $\mathcal{C} = true$ when there is no pair of examples (x_p, y_p), (x_q, y_q) such that (17) holds.

5. Strategies for Heuristic Data Driven Control of the Parameters. Experimental Analysis

The developments in this section analyze the effects of different choices of the bias parameter b^* on the generalization capacities of the resulted non-linear soft margin SVM classifier.

As it is presented in Section 4, the learning rule (12) involves the displacement parameter

$$\Delta = \rho \left(\rho_1 \frac{\partial Q(\alpha)}{\partial \alpha_{p_1}}\bigg|_{\alpha=\alpha^{old}} + (1-\rho_1) \frac{\partial Q(\alpha)}{\partial \alpha_{p_2}}\bigg|_{\alpha=\alpha^{old}} \right) \qquad (19)$$

The learning rate $\rho > 0$ and the weight $\rho_1 \in [0,1]$ should be taken such that the search process is optimized from both point of views, accuracy and efficiency. In order to obtain good approximations of the maxima values of the objective function, in our tests we used $\rho \in [10^{-4}, 1]$.

In our approach, the weight parameter ρ_1 is computed from data as follows. Let $\hat{\mu}_{g,1}, \hat{\mu}_{g,2}, \ddot{\Sigma}_{g,1}, \ddot{\Sigma}_{g,2}$ be the sample means and sample covariance matrices computed on the basis of the samples labeled by 1 and -1 in the feature space, where *g* is a particular feature extractor. We denote by *K* the kernel generated by *g*, that is $K(x,x') = g(x)^T g(x')$. Since we assumed the first *m* examples as coming from the first class and the next *N-m* examples as coming from the second class, we get,

$$\hat{\mu}_{g,1} = \frac{1}{m}\sum_{i=1}^{m} g(x_i), \quad \hat{\mu}_{g,2} = \frac{1}{N-m}\sum_{i=1}^{N-m} g(x_{m+i})$$

$$\hat{\Sigma}_{g,1} = \frac{1}{m-1}\sum_{i=1}^{m} \left(g(x_i) - \hat{\mu}_{g,1}\right)\left(g(x_i) - \hat{\mu}_{g,1}\right)^T$$

$$\hat{\Sigma}_{g,2} = \frac{1}{N-m-1}\sum_{i=1}^{N-m} \left(g(x_{m+i}) - \hat{\mu}_{g,2}\right)\left(g(x_{m+i}) - \hat{\mu}_{g,2}\right)^T$$

In (Cocianu, State, 2013) we proposed a new expression of the weight parameter $\rho_1$computed in terms of first and second order sample statistics $\rho_1 = \frac{\theta_{g,2}}{\theta_{g,1}+\theta_{g,2}}$, where $\theta_{g,i} = \frac{\left(\hat{\mu}_{g,i}^T \hat{\mu}_{g,i}\right)^2}{\hat{\mu}_{g,i}^T \hat{\Sigma}_{g,i} \hat{\mu}_{g,i}}, i = 1,2$.

Note that the expression of ρ_1 is inspired from arguments coming from mathematical statistics, based on Fisher information coefficient. Moreover, it can be computed in terms of the values of the kernel K as follows

$$\rho_1 = \frac{(N-m)^2(N-m-1)\frac{\left\|\hat{\mu}_{g,2}\right\|^4}{\sum_{k=1}^{N-m}\left(S(k)\right)^2}}{m^2(m-1)\frac{\left\|\hat{\mu}_{g,1}\right\|^4}{\sum_{k=1}^{m}\left(T(k)\right)^2} + (N-m)^2(N-m-1)\frac{\left\|\hat{\mu}_{g,2}\right\|^4}{\sum_{k=1}^{N-m}\left(S(k)\right)^2}}$$

where

$$T(k) = \sum_{i=1}^{m}\left(K(x_i,x_k) - \frac{1}{m}\sum_{p=1}^{m}K(x_i,x_p)\right)$$

$$S(k) = \sum_{i=1}^{N-m}\left(K(x_{m+i},x_{m+k}) - \frac{1}{N-m}\sum_{p=1}^{N-m}K(x_{m+i},x_{m+p})\right)$$

(Cocianu, State, 2013).

It is well known that the value of the bias parameter b^* cannot be computed by solving the QP-problem (8) and there have been proposed several computation rules expressed in terms of the support vectors, as for instance (4). In our developments we used the expression (20) proposed in (An, Liang, 2013) and we introduced the expressions (21) and (22),

where $b_i = y_i - \sum_{j=1}^{N} \propto_j^* y_j K(x_i,x_j), i = 1, \ldots, N$, and SV is the set of support vectors, in order to refine the bias by taking into account the relative importance of the support vectors.

$$b_1^* = \frac{1}{|SV|}\sum_{x_i \in SV} \propto_i b_i \qquad (20)$$

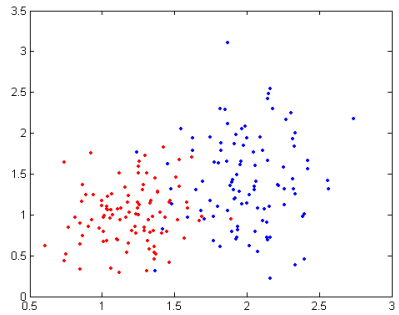
$$b_2^* = \frac{1}{\sum_{x_i \in SV} \alpha_i} \sum_{x_i \in SV} \alpha_i \, b_i \tag{21}$$

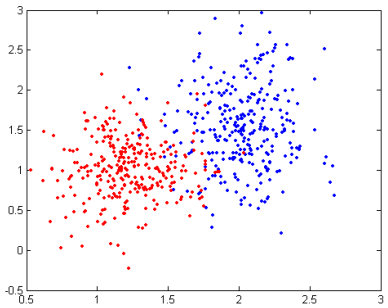
$$b_3^* = \frac{1}{\sum_{x_i \in SV} \frac{1}{\alpha_i}} \sum_{x_i \in SV} \frac{1}{\alpha_i} b_i \tag{22}$$

The effects of different choices of the bias b^* on the quality of the resulted classifier have been evaluated on experimental basis, and the results of these comparative analysis are presented in the following.

Our tests were performed on artificially generated data from Gaussian repartitions, and on Ripley and Monk's databases (http://archive.ics.uci.edu/ml/index.html). Also, we used two types of kernels, GRBF and ERBF, where the value of the parameter γ was determined such that the recognition rate is optimized. The comparative analysis of the resulted gradient ascent algorithm (GRAD algorithm) was performed against the implementation of the variant of Platt's SMO presented in Section 4.2, the linear and the quadratic discriminant function classifiers respectively, and the classifier based on Mahalanobis-type discriminant function supplied by MATLAB.

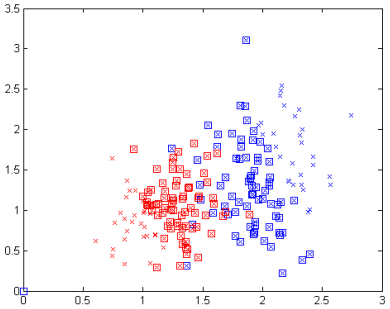
Test 1. The first series of tests were performed on data of different sizes generated from two Gaussian repartitions. For instance, the results of one of these tests performed on data generated from bi-dimensional Gaussian repartitions, where the design data consisted of 100 examples coming from each class and the test data contained 300 examples coming from each class are shown in Figure 1. The resulted support vectors in cases the kernels GRBF and ERBF respectively were used, are presented in Figure 2. The results of the comparative analysis in case the data were generated from $N\left(\binom{2}{1.5}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.3 \end{pmatrix}\right)$ and $N\left(\binom{0.5}{1}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.2 \end{pmatrix}\right)$ are summarized in Table1. The best recognition rate 95% was obtained in case of the GRAD algorithm with ERBF kernel $\gamma = 0.03$ and the bias set to either to b_3^* or b_2^*, the approximation of the solution being computed in 378 iterations.

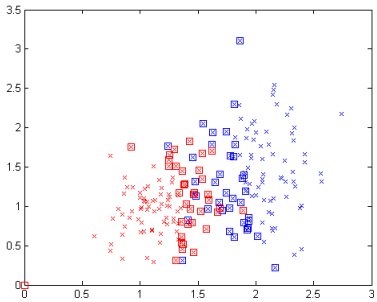




The training dataset **The test dataset**

Figure 1





The support vectors in case of GRBF **The support vectors in case of ERBF**

Figure 2

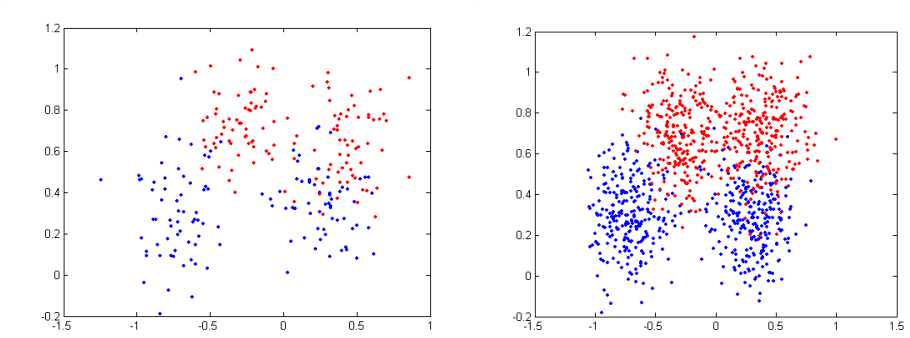
Table 1

Method	**Recognition rate**
Linear discriminant function	93.83%
Quadratic discriminant function	94%
Mahalanobis-type discriminant function	93.50%

Method	**Recognition rate**	**Number of iterations**	**Bias**	γ
Soft margin SVM SMO algorithm Gauss kernel	94.33%	780	b_1^*, b_2^*, b_3^*	0.01
Soft margin SVM	94.83%	549	b_3^*	0.01

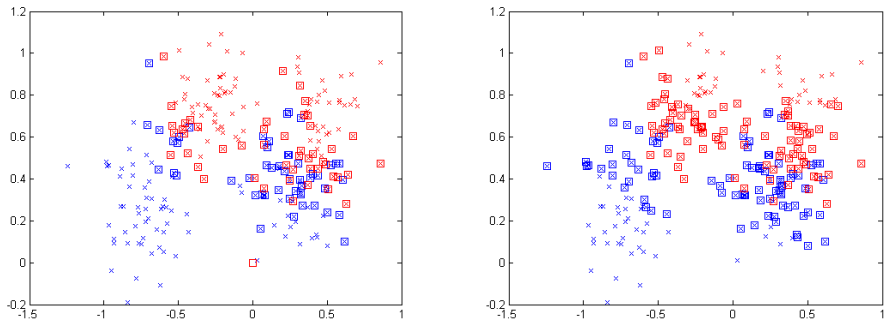
GRAD algorithm Gauss kernel				
Soft margin SVM SMO algorithm Exponential kernel	94.67%	1020	b_1^*	0.05
Soft margin SVM GRAD algorithm Exponential kernel	*95%*	*378*	b_2^*, b_3^*	*0.03*

Test2. The Ripley dataset consists of 1250 bi-dimensional samples containing data coming from two classes. The training set contains 125 samples and the testing set contains 500 samples coming from each class. The training and test sets are shown in Figure 3 and the support vectors computed by GRBF and ERBF respectively are depicted in Figure 4. The best performance was obtained by GRAD algorithm using the ERBF kernel, $\gamma = 0.01$ and bias was set either to b_3^* or b_2^*, the approximation of the solution being computed in 457 iterations yielding to the recognition rate 91.40%. The results are summarized in Table 2.



The training dataset **The test dataset**

Figure 3



The support vectors using the Gauss kernel **The support vectors using the exponential kernel**

Figure 4

Table 2

Method	**Recognition rate**
Linear discriminant function	89.20%
Quadratic discriminant function	89.80%
Mahalanobis-type discriminant function	89.10%

Method	**Recognition rate**	**Number of iterations**	**Bias**	γ
Soft margin SVM SMO algorithm Gauss kernel	89.60%	1512	b_2^*	0.2
Soft margin SVM GRAD algorithm Gauss kernel	89.90%	425	b_2^*	0.2
Soft margin SVM SMO algorithm Exponential kernel	91.40%	1064	b_3^*	0.01
Soft margin SVM GRAD algorithm Exponential kernel	*91.40%*	*457*	b_2^*, b_3^*	*0.01*

Test 3. The MONK's dataset 3 (with noise)

The MONK's dataset 3 is derived from a domain in which each training example is represented by six discrete-valued attributes. Experiments were performed on data resulted by superimposing noise on the training examples. The training data contains 62, 60 examples respectively coming from the two classes and the test data is represented by a collection of 204 examples coming from one class and 228 examples coming from the other class. The results are summarized in Table 3. The best recognition rate 94.44% was achieved by GRAD algorithm when the ERBF kernel $\gamma = 1.5$ was used and the bias was b_2^*. However, it seems that the GRAD algorithm should be preferred because it computes a good approximation of the solution in a far less number of iterations.

Table 3

Method	**Recognition rate**
Linear discriminant function	80.90%
Quadratic discriminant function	89.02%
Mahalanobis-type discriminant function	87.32%

Method	**Recognition rate**	**Number of iterations**	**Bias**	γ
Soft margin SVM SMO algorithm Gauss kernel	91.44%	845	b_1^*, b_2^*, b_3^*	1.5
Soft margin SVM GRAD algorithm Gauss kernel	91.76%	195	b_2^*	1.5
Soft margin SVM SMO algorithm Exponential kernel	94.21%	207	b_2^*	1.5
Soft margin SVM GRAD algorithm Exponential kernel	*94.44%*	*86*	b_2^*	*1.5*

6. Conclusions and Suggestion for Further Work

In the paper we propose a modified gradient ascent method for solving the dual problem of nonlinear soft margin SVM together with two new expressions of the bias.

The proposed variant of the gradient ascent learning algorithm is somehow heuristically justified in the sense that there is no mathematically founded proof of the convergence properties. Therefore, several tests were performed in order to derive conclusions on experimental basis. The tests pointed out good convergence properties and, moreover, the proposed modified variants proved higher convergence rates as compared to the Platt's SMO algorithm. The experimental analysis aimed to derive conclusions on the recognition rate as well as on the generalization capacities.

The comparative analysis was developed in terms of a long series of tests performed on artificially generated data as well as on standard databases. In Section 5 are summarized the results obtained on artificial data generated from multivariate Gaussian repartitions and on the Ripley and Monk's databases (http://archive.ics.uci.edu/ml/index.html).

The tests pointed out that the variation of the recognition rates depends also on the inner structure of the classes from which the learning data come as well as on their separability degree. Consequently, the results are encouraging and entail future work toward extending these refinements to multi-class classification problems and approaches in a fuzzy-based framework.

Acknowledgement

This paper was co-financed from the European Social Fund, through the Sectorial Operational Programme Human Resources Development 2007-2013, project number POSDRU/159/1.5/S/138907 "Excellence in scientific interdisciplinary research, doctoral and postdoctoral, in the economic, social and medical fields -EXCELIS", coordinator The Bucharest University of Economic Studies.

REFERENCES

[1]Abe, S. (2010), ***Support Vector Machines for Pattern Classification***;
In: *Advances in Pattern Recognition*, 2010, pp. 1-473;

[2]**Cocianu, C., L. State (2013),** ***Kernel-Based Methods for Learning Non-Linear SVM;*** *Economic Computation and Economic Cybernetics Studies and Research*, Volume 47, *ASE Publishing;* Issue 1, pp. 41-60;

[3]**Cortez, C., V. Vapnik (1995),** ***Support Vector Networks*** ; Machine Learning 20, 273-297;

[4] **Ho Y.-C. and Kashyap R.L. (1965),** ***An Algorithm for Linear Inequalities and its Applications****; IEEE Trans. Elec. Comp*., Vol. 14, No. 5, pp. 683–688;

[5]**Ho Y.-C. and Kashyap R.L. (1966),** ***A Class of Iterative Procedures for Linear Inequalities*** *; SIAM J. Control*., Vol. 4, No. 2, pp. 112–115;

[6]**Keerthi, S.S. and S. K. Shevade (2003),** ***SMO Algorithm for Least Squares SVM Formulations****; Neural Computation*, Vol. 15, pp. 487-507;
[7]**Levenberg, K. (1944),** ***A Method for the Solution of Certain Non-Linear Problems in Least Squares***. *Quarterly of Applied Mathematics* **2**: 164–168;
[8]**Marquardt, D. (1963),** ***An Algorithm for Least-Squares Estimation of Nonlinear Parameters***. *SIAM Journal on Applied Mathematics* **11** (2): 431–441;
[9]**Mercer, J. (1908),** ***Functions of Positive and Negative Type and their Connection with the Theory of Integral Equations***; *Proc. Roy. Soc. London Ser. A* , 83, 1908, pp. 69–70;
[10]**Osuna, E., R. Freund and F. Girosi (1997),** ***An Improved Training Algorithm for Support Vector Machines*** ; In *Proc. the IEEE Workshop. Neural Networks for Signal Processing*, pp. 276-285;
[11]**Platt, J. (1998),** ***Fast Training of Support Vector Machines Using Sequential Minimal Optimization***; Advances in Kernel Methods – Support Vector Learning, *Cambridge, MA, MIT Press*;
[12]**Rosenblatt, F. (1957),** ***The Perceptron – A Perceiving and Recognizing Automaton****;* Report 85-460-1, Cornell Aeronautical Laboratory;
[13]**Shawe-Taylor, J. and N. Cristianini (2000),** ***Support Vector Machines and other Kernel-based Learning Methods****; Cambridge University Press*, UK;
[14]**Shawe-Taylor , J. and N. Cristianini (2004),** ***Kernel Methods for Pattern Analysis****; Cambridge University Press*;
[15]**State, L., C. Cocianu, P. Vlamos (2011),** ***A New Learning Algorithm of SVM from Linear Separable Samples*** *;* Applied Mechanics and Materials, Volume: 58-60, Pages: 983-988;
[16]**Tilman Knebel, Sepp Hochreiter, Klaus Obermayer (2008),** ***An SMO Algorithm for the Potential Support Vector Machine***; Neural Computation 20(1): 271-287;
[17]**Vapnik, V. (1995),** ***The Nature of Statistical Learning Theory*** ; *Springer-Verlag*, New York;
[18] **Vapnik, V. (1998),** ***Statistical Learning Theory***; *John Wiley*, New York;
[19]**W. Liu, J. Principe and S. Haykin (2010),** ***Kernel Adaptive Filtering: A Comprehensive Introduction*** ; *Wiley*;
[20]**W.J An, MG Liang(2013),** ***Fuzzy Support Vector Machine Based on within-class Scatter for Classification Problems with Outliers or Noises;*** *Neuro-computing*, Volume 110, pp 101-110;
[21]http://archive.ics.uci.edu/ml/index.html.