Biocybernetics and Biomedical Engineering
2009, Volume 29, Number 4, pp. 63–81

Performance of the Support Vector Machines for Medical Classification Problems

MAŁGORZATA ĆWIKLIŃSKA-JURKOWSKA*

Department Theoretical Backgrounds of Medical Sciences and Medical Informatics, Collegium Medicum, Nicolaus Copernicus University, Bydgoszcz, Poland

In the Support Vector Machines classification technique the best possible discriminating hyperplane between two populations is looked for by maximizing of margin between the populations' closest points. This idea is also applied for obtaining nonlinear discriminant boundaries by using different kernels for transformations, thus obtaining a nonlinear Support Vector Machines method. The nonlinear Support Vector Machines method is based on pre--processing of data to represent patterns in high dimension- usually much higher than the original variable feature space.
In the presented work the dependency of Support Vector Machines performance on the kind of kernel and Support Vector Machines parameters is presented. The performance was assessed by resubstitution, 10-fold cross-validation, leave-one-out error, learning curves and Receiver Operating Characteristic curves. The kind and shape of the kernel is more important than regularization constant allowing different levels of overlapping classes. Combining boosting and Support Vector Machines did not improved performance in comparison to Support Vector Machines method alone, because both Support Vector Machines procedure and boosting are focused on observations difficult to classify.

K e y w o r d s: Support Vector Machines, regularizing constant, kernel function, kernel parameter selection

1. Introduction

There is a variety of methods for linear discrimination in the two-class case: Fisher linear discrimination, least mean squared error-pseudo-inverse, perceptron, relaxation and Support Vector Machines. Some of them can find a boundary that divides

* Correspondence to: M. Ćwiklińska-Jurkowska, Dept. Theoretical Backgrounds of Medical Sciences, Collegium Medicum, Nicolaus Copernicus University, ul. Jagiellońska 13, 85-067 Bydgoszcz, Poland, e-mail mjurkowska@cm.umk.pl
Received 04 December 2008; accepted 30 April 2009

two classes if they are separable, others can not. The Support Vector Machines method [1] provides an optimally separating hyperplane in the sense that the margin between two groups is maximized. Expanding of this idea to the nonlinear classifier provides classifiers with significantly good generalization properties. The Support Vector Machines classification technique is based on mapping the data to represent observations in high dimensional space- usually much higher than the original feature space. With an appropriate nonlinear transformation to a sufficiently high dimension, data from two categories can always be separated by a hyperplane. The main advantage of the Support Vector Machines is that complexity of the classifier is determined by the number of support vectors rather than the dimensionality of the transformed space. As a consequence, Support Vector Machines have less often problems with overfitting than many other methods. User of Support Vector Machines can avoid overfitting by controlling the margins. Support Vector Machines method minimizes the expected generalization error rather than apparent error. Theoretical bounds on the expected generalization error are given. Another benefit is the global optimization (solution of the optimization problem is the global minimum). More, for Support Vector Machines the "curse of dimensionality" is avoided- Support Vector Machines is appropriate for high dimensionality. A great benefit of Support Vector Machines results also from the simple implementation coming from kernel formulation. On the other hand, a drawback of Support Vector Machines is long training time for large sets, because quadratic optimization scales poorly with the number of training examples.

Kernel functions can range from simple linear and polynomial transformations to sigmoidal kernels and radial basis functions. Various kernel functions have different properties and different kind of parameters, apart from the common regularizing (penalty) parameter C. Support Vector Machines procedure has only few free parameters. Thus it is interesting to study importance of the different parameters connected with the kernel function. Importance of the kernel shape and the kernel parameter selection for good performance is very important topic. In model selection the user should make a decision which one of the universal kernel functions (e.g. linear, polynomial, radial basis and sigmoidal) will be examined. After that, the parameters of Support Vector Machines typically must be selected to give reasonable results. Good parameter selection is fundamental to the Support Vector Machines' success. The kernel function settings are hard to select before seeing the training data, so experimental methods will be used for choosing of kernel parameters. The clear method is to estimate an array of possible settings, for example the regularizing parameter C (penalty for overlapping), the polynomial order and other kernel parameters. By putting some error assessment (leave-one-out or more general cross-validation) into this table the best setting of parameters can be discovered.

The aim of the work is to study the Support Vector Machines' performance for various kernels (Gaussian, polynomial, radial sigmoidal, and linear), different kernel

parameters and different dimensionalities on the basis of medical data sets and to check usefulness of boosting combined with Support Vector Machines.

Calculations were done by own programs using Matlab6 and the svc procedure from PRTOOLS4 package and by the svc procedure from package e1071 of R .

2. Support Vector Machines

Linear functions in discrimination can be divided in methods performed in the original feature space (e.g. linear discriminant function, linear perceptron, linear Support Vector Machine) and in the transformed feature space (nonlinear Support Vector Machine, other kernel methods). Thus Support Vector Machines are linear methods in the original or transformed feature spaces.

Support Vector Machines up till now represent a powerful technique for nonlinear classification, regression and outlier finding. Support Vector Machines were developed by Cortes & Vapnik [1] from Vapnik's work on Structural Risk Minimization. The Support Vector Machines method was elaborated in the machine learning theory for binary classification. The interesting feature of Support Vector Machines method is an intuitive model representation. The approach may be generally characterized as follows: the best possible discriminating hyperplane between the populations is looked for by maximizing the margin between the populations' closest points. The *margin* is a minimal Euclidean distance between any training example and the separating hyperplane. The observations which are lying on the boundaries (support hyperplanes) are called the "support vectors" and the middle between the support hyperplanes is the optimally separating hyperplane. The support vectors are the training samples that identify the optimal separating hyperplane and are the most difficult to classify.

First of all the simplest linear Support Vector Machines will be presented. For separable data the algorithm is defined as follows:

The equation for a separating hyperplane in the linear Support Vector Machines method is

$$f(x) = \boldsymbol{w}^T \boldsymbol{x} + b \tag{1}$$

where vector $\boldsymbol{w}$ is normalized such that:

$$\min_{[i = 1,..,n]} |f(x_i)| = 1 \text{ holds}$$

and n- is the number of observations in learning set.
For x which is the support vector (SV) we have

$$\boldsymbol{w}^T \boldsymbol{x} + b = 1 \text{ or } \boldsymbol{w}^T \boldsymbol{x} + b = -1.$$

Then the margin is equal to $2/||\boldsymbol{w}||$. The value $1/||\boldsymbol{w}||$ is the distance of the support vectors from the decision hyperplane. The larger the margin the lesser the generalization error of the Support Vector Machines classifier.

The Support Vector Machines method finds the separating hyperplane with the largest margin, so dual optimization problem for Support Vector Machines is:

– We have the objective function

$$\min_{[\mathbf{w},b]} 1/2\|\mathbf{w}\|^2$$

– subjected to constrains

$$y_i[\mathbf{w}^T\mathbf{x} + b] \geq 1 \qquad \text{for } i = 1,..,n \tag{2}$$

where $y_i = 1$ or $y_i = -1$.

The optimization problem is quadratic-linear (the task of quadratic objective function with linear constrains) and can be solved by Lagrange multiplier method. The support vectors are the training patterns for which the above inequality represents the equality so they lay on the support hyperplanes. The separating hyperplane is parallel to the support hyperplanes and is laying in the middle of distance between the support hyperplanes.

The support vectors are the training observations that identify the optimal separating hyperplane. In contrast to many neural networks methods one can always find the global minimum, although minimum may be not unique as for example in the case when dimensionality of the problem d is smaller than n (number of observations). Vapnik-Chervonenkis (VC) dimension of a set of functions is defined as a maximum number of training points that can be shattered i.e. separated for all possible labeling. If the VC dimension is m, then there is at least one set of m points that can be shattered, but not necessarily every set of m points can be shattered. For linear discriminant function it is equal to $d + 1$, where d is the dimensionality.

Vapnik [2] shown that the value v which is the Vapnik-Chervonenkis dimension VC for the linear Support Vector Machines is bounded by:

$$v \leq min\,(\Lambda^2 R^2 + 1,\ d + 1), \tag{3}$$

where the equation for the boundary hyperplane is $f(x) = \boldsymbol{w}^T\mathbf{x} + b$, the dimensionality of the problem is d and Λ is such value that $||w|| \leq \Lambda$ and finally R is the radius of the smallest sphere around the data.

Large margin (margin is equal to $2/||w||$) diminishes complexity of a linear Support Vector Machines classifier. On the other hand, if we agree to smaller margins, there is bigger number of separating hyperplanes, what means that the Vapnik-Chervonenkis dimension v is bigger.

From the above inequality (3) it is clear that complexity v only indirectly depends on dimensionality of the data d, while other machine learning procedures like neural networks and classification trees depend strongly of dimensionality. There is also important difference to e.g. kernel Bayesian discrimination where the classification becomes difficult for big dimensionality of data (for the kernel discrimination the procedure of local smoothing is impossible for high dimensional data when data is sparse in such high dimensional space while for Support Vector Machines the way of using the kernel functions is quite different, so the Support Vector Machines method in opposite to the kernel discrimination is appropriate for solving multidimensional problems). For the Support Vector Machines the classification problem remains easy if a large margin can be achieved, however, the selection of the kernel is important. Support Vector Machines are not faced with the curse of dimensionality. Complexity of Support Vector Machines is connected with the number of support vectors, which are the most difficult patterns from learning data to classify.

When the classes are not linearly separable, a variant of Support Vector Machines, called a *soft-margin Support Vector Machine* is used. This variant penalizes its classification errors and employs a parameter (the soft margin constant C) to control the cost of misclassification. Parameter C is nonnegative.
For non-separable data we may relax inequality (hard-margin constrains)

$$y_i [\boldsymbol{w}^T \boldsymbol{x} + b] \geq 1$$

where $y_i = -1$ or $y_i = -1$

to become

$$y_i[\boldsymbol{w}^T \boldsymbol{x} + b] \geq 1 + K_i \quad \text{where} \quad y_i = 1 \text{ or } y_i = -1 \tag{4}$$

where $K_i \geq 0$ are called "slack" variables .
For non-separable case the modified objective function is used

$$min_{[\boldsymbol{w},b,\boldsymbol{K}]} \tfrac{1}{2} ||\boldsymbol{w}||^2 + C (K_1 + ... + K_n) \tag{5}$$

where $\boldsymbol{K} = (K_1, ..., K_n)$.

The Support Vector Machines solution can be found by keeping the upper bound on the VC-dimension small and by minimizing the upper bound of the empirical risk $K_1 + ... + K_n$. The default value of C is 1. If $C < 1$, we agree with overlapping. The smaller C the bigger overlapping. Constant C determines the trade-off between the empirical error and the complexity, so C is called a regularization constant. It is clear that the value $K_1 + ... + K_n$ is connected with the penalty for the overlapping.

The selection of constant C can be used by the user by examining of test or cross-validation error or other assessment of probability of classification error.

Choosing the constant C corresponds to regularization, because smaller values can help to avoid over-fitting of the discriminating hyperplane. It is especially important for hypersurface of more complex shape. When two classes are unbalanced, different penalty for misclassification can be associated to each of the classes. It can be realized in Support Vector Machines by means of two C parameters (by weighting C we obtain C_1 and C_2 for classes Π_1 and Π_2, respectively).

Thus support vectors are the "most informative" for the classification task and the support vectors are (equally) close to the hyperplane. They are close to the classification boundary or misclassified.

Structural Risk Minimization principle from Vapnik [2] lays ground for the Support Vector algorithm and an upper bound on the generalization error is minimized. By the use of Structural Risk Minimization the classification rule is less sensitive to the dimensionality of the space and achieves good generalization.

The computational cost of Support Vector Machines is $O(n^2 n_s)$ [7], due to solving a quadratic programming problem arising in the Support Vector Machines algorithm, where n_s is number of the support vectors. Number of the support vectors n_s usually increases linearly with n [8]. An important property of Support Vector Machines is that it only estimates $sign\ [P(Y=1|X=x) - ½]$ while the conditional probability of a point x being in population Π_1: $P(Y = 1|X = x)$ is often of interest.

Fundamental to the success of Support Vector Machines was re-discovery of the so-called Reproducing Kernel Hilbert Spaces and Mercer's [3] theorem (the kernels satisfying the Mercer' theorem are called Mercer kernels). The basic idea of the so-called *kernel methods* is at first preprocessing of the data by some non-linear mapping Ψ and then applying the same linear algorithm as mentioned before but in the image space of mapping Ψ. Scalar product is now definite by the positive definite kernel function K

$$K(\boldsymbol{x},\boldsymbol{x'}) = \Psi(\boldsymbol{x})^T \Psi(\boldsymbol{x'}). \tag{6}$$

The data set is mapped from the original d-dimensional space by the function Ψ induced by the kernel function K. The kernel trick from Vapnik [2] is to take the original algorithm for the linear Support Vector Machines and formulate it such, that we only use $\Psi(\boldsymbol{x})$ in scalar products. If we can efficiently calculate these scalar products, we do not need to carry out the mapping Ψ explicitly.

For the nonlinear Support Vector Machines data are nonlinearly transformed to a high-dimensional feature space. The nonlinear classification boundary in the original d-dimensional space corresponds to a linear boundary in the transformed by Ψ feature space. The dimension of the transformed space can be very large, even infinite in some cases.

After such kernel transformation we look also for linear boundaries that give optimal class separation in such transformed data spaces. Linear, polynomial, Gaussian, radial basis function, inverse multiquadratic and sigmoidal kernels are most often

used. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two groups can always be discriminated by a hyperplane, though then the method can have a drawback of overfitting.

At the moment Support Vector Machines have become a popular technique in flexible modeling, though there are some disadvantages, eg. Support Vector Machines method works relatively badly with the data size- due to the need of quadratic optimization procedure. Additionally, the correct choice of the kernel parameters is fundamental for obtaining good results, which means that a wide search must be performed on the parameter space, and this frequently makes the work difficult.

Expected value of generalization error e_G of the Support Vector Machines classifier is bounded by [2]

$$E_n\,(e_G) \le E_n\,(N_s)/n \tag{7}$$

where

n – number of patterns,

N_s – the total number of support vectors,

E_n – denotes the expectation over all training sets of size n.

The bound of $E_n(e_G)$ is independent of the dimensionality of the vectors in the transformed space determined by function Ψ. When we can find a transformation Ψ that well separates the data set (the $E_n(N_s)$ – expected number of SV is small) then from the above bound expected generalization error will be low. Thus the number of support vectors has a relationship with accuracy.

The kernel functions $K(\boldsymbol{x},\boldsymbol{y})$ can be defined by the following types:

$$\text{polynomial } K(\boldsymbol{x},\boldsymbol{y}) = (\gamma\, \boldsymbol{x}\boldsymbol{y}' + a)^p \tag{8}$$

$$\text{homogeneous } K(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}\, \boldsymbol{y}')^p \tag{9}$$

$$\text{exponential } K(\boldsymbol{x},\boldsymbol{y}) = exp(-(||\boldsymbol{x}\text{-}\boldsymbol{y}||)/p) \tag{10}$$

$$\text{radial basis } \mathrm{K}(\boldsymbol{x},\boldsymbol{y}) = exp(-(\gamma||\boldsymbol{x}\text{-}\boldsymbol{y}||/p)^2)) \tag{11}$$

$$\text{sigmoidal } K(\boldsymbol{x},\boldsymbol{y}) = s[(\boldsymbol{x}\, \boldsymbol{y}')/p]$$

where s is sigmoidal function, for example (12)

$$s(x) = \exp\,(x)/[1 + \exp(x)] \tag{13}$$

or

$$K(\mathbf{x},\mathbf{y}) = \tanh\,(\,\gamma\, \mathbf{x}\, \mathbf{y}' + a). \tag{14}$$

If s is sigmoidal kernel function then
$\lim_{t\to -\infty} s(t) = 0$ and $\lim_{t\to\infty} s(t) = 1$.

Easy, symmetric functions of two vector arguments from the original d-dimensional space R^d called kernel functions allow to count the scalar product in the transformed space.

Chang & Lin [4], the authors of the "libsvm" procedure in R package, performed some work on methods of efficient automatic parameter selection. The existing

implementation is optimized for the radial basis function kernel only, which obviously might be suboptimal for some data sets.

Many authors have noted the relationship between Support Vector Machines and regularized function estimation in the reproducing kernel Hilbert spaces (e.g. [5]).

However the differences of Support Vector Machines with the kernel methods may be underlined:

– For suitable chosen transformation Support Vector Machines method has a powerful nonlinearities but still is very intuitive,
– Support Vector Machines procedure retains most of the favorable properties of its linear input space version.

Similarity of Support Vector Machines by mapping data can be also noticed for the following methods:

– neural network (single hidden layer): input data are mapped to some representation given by a hidden layer,
– radial basis neural network (RBF bumps),
– boosting algorithm.

The Support Vector Machines can be extended from the two-class classification to the multiclass case [2, 6]. Multi-class generalizations are obtained by combining of two-class Support Vector Machines procedures results. Well-known aggregation methods of two-class base classifiers are one-against-one class and one-against-all classes. Then the problem is changed by k Support Vector Machines two-class rules, which can be combined by different procedures.

As it was written above, the computational cost of Support Vector Machines is $O(n^2 n_s)$, where n_s is number of the support vectors. To reduce the computational costs, diminishing of the number of essential points used in the procedure was proposed by Zhu and Hastie [7], who obtained cost of $O(n^2 m^2)$, where m is number of the "important points". The Import Vector Machine method is faster than Support Vector Machine, because the number of important points m is as a rule much smaller than the number of support vectors n_s (m does not tend to increase as n increases).

One extension of Support Vector Machines is that for the regression task, another is one-class classification. Bennett & Campbell [8] gave an overview of Support Vector Machines. On the contrary to some papers other authors concluded that boosting of Support Vector Machines is not beneficial for performance. However, it is well known that boosting of Support Vector Machines is very time-consuming.

Support Vector Machines perform well in pattern recognition, test classification and bioinformatics. Biological experiments from laboratory technologies like microarray and proteomic techniques create data with a very high number of variables, in general much larger than the number of examples. For that reason the feature selection gives an essential step in the analysis of such type of data. The feature selection in proteomic pattern data with Support Vector Machines can be done by a recurrent

feature elimination (RFE) and a recurrent feature replacement (RFR). The recurrent feature elimination method can be used for finding of starting gene subsets in the recurrent feature replacement method.

Performance of the Support Vector Machines methods for different kernels and different kernel parameters was compared on the ground of cross-validation, leave-one-out error (which is a special case of cross-validation for number of folds equal to the size of training sample), learning curves and Receiver Operating Characteristic (ROC) curves.

3. Data Sets

Medical data from the UCI Repository of Machine Learning Databases (http://archive.ics.uci.edu) giving the different dimensions were applied to the classification. Mainly results for hepatitis data (155 patients), consisting of two classes: dead and alive, have been presented. Patients are characterized by 19 clinical features. This data set comes from G. Gong form Carnegie-Mellon University.

Additional two data sets (diagnostic and prognostic) connected with breast cancer from the University of Wisconsin are also considered. Wisconsin Diagnostic Breast Cancer (*WDBC*) data set consists of 569 patients divided into two groups (malignant, benign) and 30 variables based on ten real-valued features which are computed for each cell nucleus (radius, texture, perimeter, area, smoothness, concavity, concave points, symmetry and fractal dimension). The variables are computed from a digitized image of a fine needle aspirate of a breast mass. The data set does not contain missing values.

Wisconsin Prognostic Breast Cancer (*WPBC*) data set consists of 198 instances described by 31 real-valued variables and one quantitative with 4 missing values. They describe characteristics of the cell nuclei present in the image, like mean, standard deviation and worst or largest value of characteristics based on ten real-valued features which are computed for each cell nucleus (radius, texture, perimeter, area, smoothness, concavity, concave points, symmetry and fractal dimension). Classification variable is 2-year recurrence of breast cancer (151 patients with and 47 patients without recurrence).

4. Results and Discussion

For analysis of hepatitis data two types of dimensionalities were considered: two-dimensional discrimination with visualization of the resulting support vectors (where the selected most discriminating variables by minimizing 1-nearest neighbor 10-fold cross-validation error for *Hepatitis* data set are: bilirubine and prothrombine time) and the whole 19-dimensional feature space.

The function to map the original to the higher-dimensional space is selected on basis of the designer understanding of the research area or from some class of kernels. For example, the most popular kernels may be considered: Gaussian, radial, sigmoidal kernels and polynomial or linear ones.

Different Support Vector Machines kernels and different values of parameters C are studied (C not greater and also bigger than 1). Parameter C allows for overlapping groups if it is smaller than 1. This parameter is connected with the penalization classification errors. Also the power (p) of raising the base kernel is considered (e.g. for a linear kernel, which after transforming with parameter $p = 2$ gives the quadratic kernel; higher values of p are equivalent to applying of the high-degree polynomial kernels).

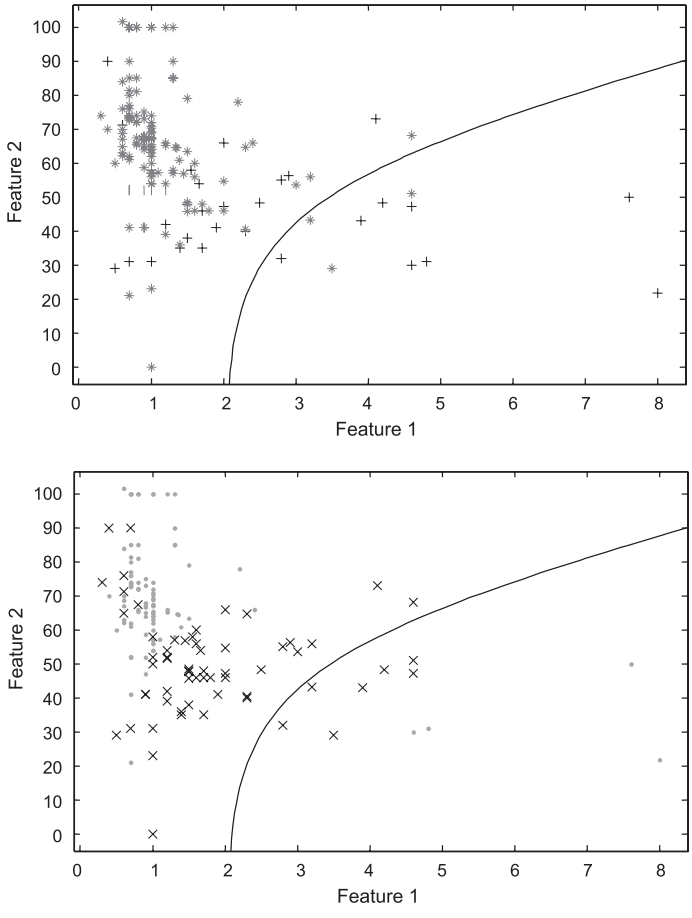

Fig. 1. Classification boundary of the quadratic Support Vector Machines discrimination between two classes (upper, "*" and "+") with the corresponding support vectors (down, "x") for Hepatitis data set. Regularization parameter $C = 1$. Feature1-bilirubine, feature 2 – prothrombine time. Percentage of SVs = 0.37. Apparent error = 0.17. Leave-one-out = 0.19

Additionally, two different dimensionalities of discrimination are examined. First, the two-dimensional discrimination is performed in order to illustrate the performance for different kernels and different regularization parameters *C* with visualization of the resulting support vectors on the plane. The most discriminating variables are chosen by iteratively selecting two optimal features using the criterion of 1-Nearest Neighbor error. The selected variables are: bilirubine (Feature 1) and prothrombine time (Feature 2).

Figures 1–2 plot the classification boundaries for polynomial with degree $p = 2$ (quadratic*)* and the radial kernels and for different parameters *C*. Upper parts of the plots in Fig. 1 consist of points denoted by "*" and "+" depending on the group. Sign "+" denotes death and "*" denotes "patient alive". On the corresponding down part of each plot "*x*" denotes the support vectors, the remaining points represent those observations that are not support vectors.

For quadratic kernel function (polynomial with power $p = 2$) with regularization parameter $C = 1$ the obtained results are presented in Fig. 1. Percentage of the support vectors relative to the size of learning sample is equal to 37%, resubstitution error is 0.17 and leave-one-out error is 0.19. For considerable changed regularization parameter $C = 0.6$ the figure very similar as presented in Fig. 1 was obtained with only the slightly higher direction of the boundary quadratic line and the slightly higher resubstitution error equal 0.18 with percentage of the support vectors and the leave-one-error remaining the same. The results repeat for many values of the changed *C* in (0,1).

Figure 2 represents a radial kernel Support Vector Machines with parameter $C = 0.85$ (resubstitution error equals 0.09, leave-one-out error equals 0.23 and

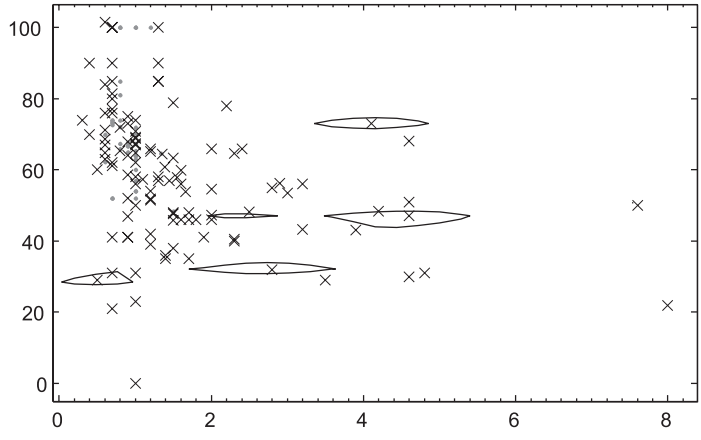

Fig. 2. Classification boundary of the radial Support Vector Machines discrimination between two classes with the corresponding support vectors („x") for Hepatitis data set. Regularization parameter $C = 0.85$. Feature1-bilirubine, feature 2 – prothrombine time. Percentage of $SVs = 0.7$. Apparent error = 0.09. Leave-one-out = 0.23

percentage of SVs is 70%). For changed values of C very similar results are obtained. For example, after choosing of parameter C equal to 0.95 the results of percentage of SVs is a slightly higher (72%), however the resubstitution error remains of the same value 0.09 and the leave-one-out assessment of error is again equal to 0.23.

The plots illustrate under- and overfitting. The simple quadratic function (Fig. 1) may underfit the data and makes a learning error. The complex nonlinear radial kernel Support Vector Machines function (Fig. 2) has evident slighter training error (0.09), however, it is not generalizing well on unseen data, because the leave-out error is relatively big (about 0.23). In case of the quadratic Support Vector Machines with $C = 1$ and $C = 0.6$, where the discriminant boundary is not overfitted, the leave-one-out errors (0.19) are similar to the training errors (0.17 and 0.18, respectively).

It is interesting to investigate support vectors because of the connection between Support Vector Machines and boosting methods. For linear classifiers and many different values of the regularizing parameter C (however, in Fig. 1 presented for one value of C only) percentage of the support vectors is 37% of the whole sample. In contrast, for the radial basis function kernel (also for different values of C) the fraction of support vectors is much bigger: about 70% (chosen value of C equal to is 0.85 presented in Fig. 2).

To find the best parameters (the regularizing parameter C, the power p of raising the base kernel and the other kernel parameters), the cross-validation or the leave-out error assessment can be applied. For radial basis kernels trying exponentially growing sequences of C and p is a useful method to find good parameters (for example, $C = 2^{-5}, 2^{-3}, 2^{-1}, 2, 2^3, 2^5$, $p = 2^{-5}, 2^{-3}, 2^{-1}, 2, 2^3, 2^5$) [4]. In the two-dimensional polynomial kernel model for *hepatitis* data, the linearly growing sequence of C and p will be used and leave-one-out error assessment methods are applied.

Values of the leave-one-out errors and percent of the support vectors as the function of two arguments: regularizing constant C and parameter p of power (the degree of polynomial kernel p in formula (8)) are considered. The functions are approximated by the table of the leave-one-out error (and percentage of support vectors, respectively) counted on the grid from 0 by 0.05 to 5 for C and values of power p from 1 to 10. Sample illustration of a part of the table for the quadratic kernel ($p = 2$) and values of the regularizing parameter C from the interval (0.1 > is presented in Fig. 3. The dashed line (corresponding to the left vertical axis) represents the leave-one-out error while the solid line (corresponding to the right vertical axis) is the percentage of support vectors.

All grid points of C and other kernel parameters are examined to observe which one gives the highest accuracy. From the tabulated values (the whole big table is not presented here) the following outcomes can be derived. For values of power p greater than 5 the leave-one-out error and percentage of the support vectors do not depend on value of C parameter. For p greater than 3 the leave – one-out errors are not changing for values of C greater than 0.25, though corresponding percentage of the support vectors are slightly changed. For values of p smaller than 3 the

values of *C* smaller than 1.5 have a little impact on the leave-one-out error and also corresponding percentage of the support vectors are a bit changed. Thus we can conclude that for polynomial kernel more important than the regularizing constant *C* (which makes possible to overlap the classes) is the coefficient of power (of raising) *p*. Such kind of the analysis can be applied to choose optimal parameters *C* and *p*. For presented *hepatitis* data the satisfying pair of parameters is *p* in <4,6> and any value of *C*, achieving the smallest leave-one-out error equal to 0.12. For such selected parameters *C* and *p* the corresponding fraction of the support vectors out of all training observations is above 60%.

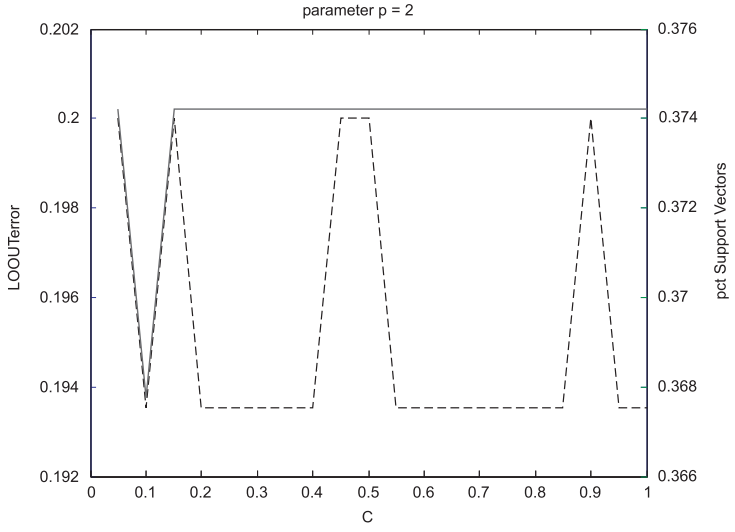

Fig. 3. Values of the leave-one-out error and percentage of the support vectors for quadratic kernel ($p = 2$) depending on values of the regularizing parameter *C* from the interval (0,1>. The dashed line – the leave-one-out error; the solid line – percentage of the support vectors. Hepatitis data set with two most discriminating variables

Next the multidimensional discrimination (19 attributes) of the hepatitis data set was examined. For many *C* and *p* values the multidimensional discrimination (19 features) gave better results than for two best discriminating variables presented on the above-discussed figures (Figures 1–3). The dependencies of the cross-validation error and percentage of the support vectors on different parameters of the Support Vector Machines method with kernels defined by formulas (8) and (11) are also studied.

Pearson coefficients of cross-validation error (and also percentage of support vectors) with regularization *C* and other Support Vector Machines parameters for hepatitis data set with all 19 variables are examined. Table 1 presents the correlation coefficients. It is a summary of the cross-validation error depending on the regularization parameter *C* and parameters of the kernel function (as power *p* of polynomial kernel and γ- the coefficient for the radial kernel- see formulas (8) and (11)).

The correlation coefficients were obtained on basis of the cross-validation error values calculated on the grid constructed as a combination of *1000* equally distanced values of the regularizing parameters *C* from the interval (0.10> and additionally for some of the following parameters. Those additional parameters depend on the kernel type. One of them is the power *p* from the set {1,...,10} (it is the degree applied for the polynomial kernel). Another examined parameter is γ. This parameter can be applied for polynomial as well as radial kernel function- see formulas (8) and (11). The values of parameter γ differ between the minimum value equal to 1/(2**p*) and the maximum value equal to 4*/p* (p-number of features). Thus kernel parameter γ is from interval <0.026, 0.21> . The last studied parameter is constant term *a* (for polynomial kernel- formula (8)). It has values from the interval <1.10>.

Table 1. Correlation coefficients of the Support Vector Machines cross-validation errors with the regularization parameters *C* and other kernel parameters. *Hepatitis* data set with 19 variables

Number	Kernel function	Correlation with C	Correlation with kernel parameter	
1	Polynomial with γ = 0.026, *a* = 0	–0.3	p	0.83
2	Polynomial with degree *p* = 2	–0.24	γ	-0.43
			a	0.13
3	Polynomial with degree p = 3	0.13	γ	0.33
			a	0.14
4	Polynomial with degree *p* = 4	–0.12	γ	–0.64
			a	0.15
5	Polynomial with degree *p* = 5	–0.04	γ	–0.37
			a	0.18
6	Polynomial with degree *p* = 6	–0.06	γ	–0.58
			a	0.28
7	Radial	–0.25	γ	0.33

Correlations of the cross-validation errors in all considered configurations of kernel parameters *C, p, γ, a* are summarized in Table 1. The dependence (assessed by absolute value of the correlation coefficient) of the cross-validation (CV-10) errors on the regularizing parameter *C* is smaller than the dependence of the cross-validation errors on other considered parameters presented in Table 1. This advantage is especially visible for the polynomial Support Vector Machines and the degree parameter *p* (from 1 to 10) i.e. correlation equal to 0.83 for dependence of CV-10 error on degree of polynomial versus the value of correlation equal to |–0.3| for dependence of CV error on the parameter *C*. Thus, for the polynomial kernel the correlation of the cross-validation error depends most strongly on the degree of polynomial (correlation

equal to 0.83). Relationships between CV errors and γ parameter of the polynomial (p from 2 to 6) and the radial kernels are smaller. The dependence of CV errors on the parameter "*a"* of the polynomial kernel function measured by correlation coefficient is below 0.2.

The selection of the Support Vector Machines (parameter *C* and also parameters of the kernel function) can be done on the basis of testing sample, cross-validation error or leave-one-out error. We should avoid too big fitting of the shape of discriminant hypersurface to data set (like too big parameter p in polynomial kernel). For example, it is not useful to choose values of p grater than 6, because the high number of the support vectors and increasing of the leave-one-out classification errors are obtained.

Table 2. Correlation coefficients of percentage of the support vectors with the regularization parameters *C* and dimensionality of ascending subsets of the most discriminating variables sorted by the criterion of Wilks lambda. *Hepatitis* data set with *19* variables

Number	Kernel function	Correlation with the regularization parameter C	Correlation with dimensionality d in the original space
1	Linear	–0.11	–0.71
2	Polynomial, degree $p = 2$	–0.36	0.79
3	Polynomial, degree $p = 3$	–0.12	0.92
4	Polynomial, degree $p = 4$	0.07	0.94
5	Polynomial, degree $p = 5$	0	0.97
6	Polynomial, degree $p = 6$	0.06	0.95
7	Radial	–0.36	0.76

Correlations of the number of the support vectors in the considered configurations of parameters *C* and *d* (dimensionality) are summarized in Table 2. For the polynomial and radial kernels the strong relationship between the ratio of the support vectors to the size of training set with the dimensionality of the data should be underlined (Table 2).

The performance of Support Vector Machines was also examined in terms of AUC- the area under Receiver Operating Characteristic curves (ROC). The latter was performed for *WPBC* and *WDBC* data.

For Wisconsin Prognostic Breast Cancer data set (WPBC) the discrimination on the whole feature space (30 variables) and additionally (to make faster the boosting Support Vector Machines procedure, which is time-consuming) for 5 most important variables chosen by the criterion of 10-fold cross-validation error for 1-nearest neighbor was examined. For Wisconsin Diagnostic Breast Cancer data set (WDBC) the discrimination based on all 32 features was studied.

For *WDBC* data set the ROC curves of radial Support Vector Machines for different values of the regularizing parameter *C* are practically identical – for all values of *C* in {1/16, 1/8 , ¼, ½, 1, 2, 4, 8, 16} and the AUC values are the same (Fig. 4). Similar situation – in comparing the ROC curves for different *C* parameters – is obtained for *WPBC* data set, which is known as difficult for classification. For example, after application of quadratic kernel (polynomial kernel with the degree equal to *2*) the ROC curves of Support Vector Machines for different values of regularizing parameters *C* (1/16, 1/8 , ¼, ½, 1, 2, 4, 8, 16) are practically identical (Fig. 5).

Combining of Support Vector Machines with the boosting method for the examined data sets was found not useful. Example result is visible in Fig. 6, which presents the learning curve (plot showing the dependency of errors from number

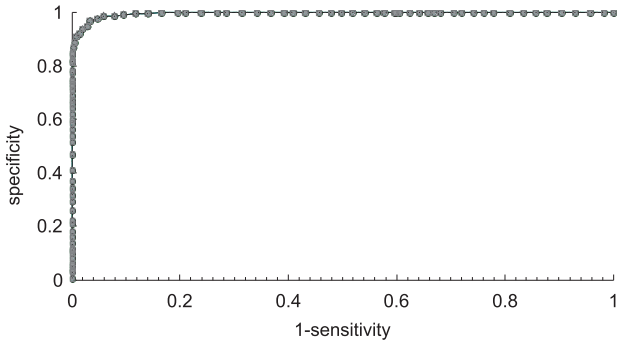

Fig. 4. Overlying ROC curves for the different regularizing parameters *C* (1/16, 1/8 , ¼, ½, 1, 2, 4, 8, 16) for Support Vector Machines with the radial kernel. WDBC data set with 30 variables

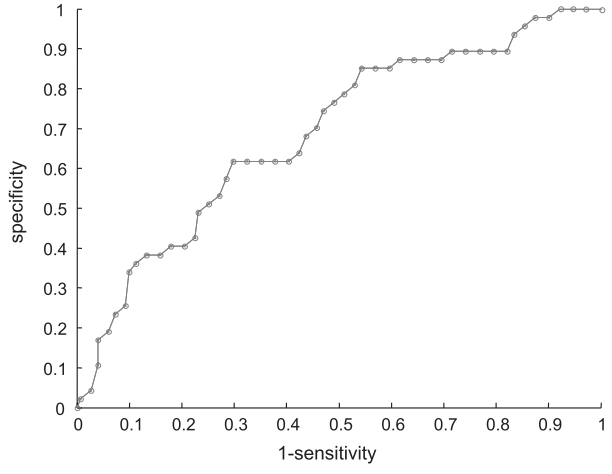

Fig. 5. Overlying ROC curves for the different regularizing parameters *C* (1/16, 1/8 , ¼, ½, 1, 2, 4, 8, 16) for Support Vector Machines with the quadratic kernel. WPBC data set with 32 variables

of loops) for 10, 20, …, 200 loops of the boosting radial Support Vector Machines base classifier (raised to the power 3). This figure presents difficult to discriminate Wisconsin Prognostic Data Set with chosen 5 most discriminating variables (selected on the basis of minimizing 10-fold cross-validation error for $k = 1$ nearest neighbor). From the learning curve it is visible that only the resubstitution error has decreased with the succeeding loops (from 10 to 200). However, no improvement by the criterion of leave-one-out and cross-validation classification error estimate was obtained after comparison with the alone base classifier (where the cross-validation and leave-one-errors are equal to 0.297). So the boosting Support Vector Machines for the considered data was highly overfitted and not helpful.

The boosting Support Vector Machines did not also improve the performance for the different data sets examined in the work. Usefulness of the boosting Support Vector Machines is not consistent by different authors. The boosting combined with the Support Vector Machines method is complex and time consuming and according to most authors it does not give improvement of performance. On he other hand, by Lili at al. [9] the ensemble of boosting with Support Vector Machines has proven to be possible beneficial, but it is too complex to be practicable. The authors set up a successful method to boost Support Vector Machine. It applies the inspiration of dynamic learning to dynamically select "important" samples into learning sample set for building base classifiers. This technique retains a small training sample set with specified size in order to control the complexity of each base classifier. In a different way than creating each base Support Vector Machines classifier directly, it uses the learning sets only for finding the support vectors. This technique to merge boosting and Support Vector Machines has been proven to be accurate and efficient by experimental outcomes.

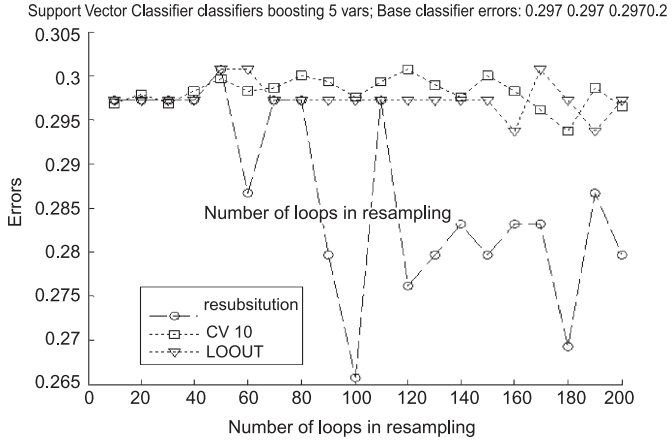

Fig. 6. Learning curve. Resubstitution, cross-validation and leave-one – out errors for combining the radial Support Vector Machines (raised to the power 3) with 10, 20,…, 200 loops of the boosting. *WPBC* data set

In SVC – global optimization in order to maximize the minimal margin is achieved, while in boosting one maximizes the margin locally for each learning observation. Thus the Support Vector Machines and the boosting are both based on maximizing of margins. Additionally in both methods idea is focused on objects difficult to classify The object obtaining large weights may occur the same as the support vectors: Skurichina and Duin [10] found that on average, support vectors found by Support Vector Machines get larger weights in the boosting procedure, than non-support vectors, however, objects with large weights in the boosting are not identical to the support vectors found by the SVC method. Those similarities of Support Vector Machines and boosting may be the reason of failure to improve the classification performance by combining boosting and Support Vector Machines method.

5. Conclusions

For the performance of Support Vector Machines more important than the regularizing parameter *C* is the kind of the kernel and additionally the power to which the kernel (for example in the polynomial kernel) is raised. Thus a shape of the kernel transformation is the most important.

Strong dependence of ratio of the support vectors to the size of training set on the dimensionality of the data can be noted.

Ensemble of Support Vector Machines with boosting gave not advance on performance by the cross validation errors criterion in comparison to the alone Support Vector Machines base classifier. This can be explained by the fact that both Support Vector Machines and boosting pay attention on observations hard for discrimination.

References

1. Cortes C. , Vapnik V.: Support-vector network. Machine Learning, 1995, 20, 1–25.
2. Vapnik V.N.: Statistical learning theory. Wiley, New York 1998.
3. Mercer J.: Functions of positive and negative type and their connection with the theory of integral equations. Philos. Trans. Royal Society London, 1909, A 209, 415–446.
4. Chang C.C. & Lin C.-J.: LIBSupport Vector Machine: a library for Support Vector Machines. 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm.ps.gz [Accessed 2009, Feb 1]
5. Hastie T., Tibshirani R. and Friedman J.: The Elements of Statistical Learning. Springer, New York 2001.
6. Lee Y, Lin Y, Wahba G.: Multicategory Support Vector Machines, Theory and Application to the Classification of Microarray Data and Satellite Radiance Data. Journal of American Statistical Association. 2004, 99. 67–81.
7. Kaufman L.: Solving the quadratic programming problem arising in Support Vector Machines; In: Advances in Kernel Methods-Support Vector Learning, eds. B. Scholkopf, C. Burges. C, Smola A., Cambridge, MA: Mit Press, 1998, 147–168.

8. Zhu J.T. Hastie T.: Kernel logistic regression and the import vector machines. Journal of Computational and Graphical Statistics, 2005, 14, 185–205
9. Bennett K.P. , Campbell C. Support vector machines: Hype or hallelujah? 2000. SIGKDD Explorations, 2(2). http://www.sigkdd.org/exploation/issue2-2/benett.pdf [Accessed 2009, Feb 1].
10. Skurichina M., Duin P.W.: Bagging, boosting and the RSM for linear classifiers. Pattern Analysis and Applications, 2002, 5, 121–135.
11. Diao L., Hu K., Lu Y., Shi C.: A method to boost Support vector Machines. Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002, Taipei, Taiwan, May 6–8, 2002. Proceedings. Editors: M.-S. Chen, P.S. Yu, B. Liu (Eds.): Springer, 2002, 463–468.