A Survey on Potential of the Support Vector Machines in Solving Classification and Regression Problems

Luminita STATE[1], Catalina COCIANU[2], Doina FUSARU[3]
[1]University of Pitesti
[2]Academy of Economic Studies, Bucharest
[3]Spiru Haret University, Bucharest
lstate@clicknet.ro, ccocianu@ase.ro, dfusaru.mfc@spiruharet.ro

Kernel methods and support vector machines have become the most popular learning from examples paradigms. Several areas of application research make use of SVM approaches as for instance hand written character recognition, text categorization, face detection, pharmaceutical data analysis and drug design. Also, adapted SVM's have been proposed for time series forecasting and in computational neuroscience as a tool for detection of symmetry when eye movement is connected with attention and visual perception. The aim of the paper is to investigate the potential of SVM's in solving classification and regression tasks as well as to analyze the computational complexity corresponding to different methodologies aiming to solve a series of afferent arising sub-problems.

***Keywords**: Support Vector Machines, Kernel-Based Methods, Supervised Learning, Regression, Classification*

1 Introduction

Support vector machines were introduced and first applied to classification problems as alternatives to multilayer neural networks. The SVM's have empirically proved to give good generalization performance on a wild variety of problems such as hand written character recognition, text categorization, face detection, pedestrian detection, pharmaceutical data analysis, and drug design. The high generalization ability provided by support vector classifiers has inspired and encouraged several attempts on computational speed ups as well as the fundamental theory of model complexity and generalization.

The main drawback in SVM based approaches is that the training algorithms for SVM's are slow, complex, subtle and difficult to implement. Training a SVM corresponds to solving a linearly constrained quadratic problem (QP) in a number of variables equal to the number of data points, this optimization problem becoming challenging when the number of data points exceeds few thousands. Because the computational complexity of the existing algorithms is extremely large in case of few thousands support vectors and therefore the SVM QP-problem becomes intractable, several decomposition algorithms that do not make assumptions on the expected number of support vectors have been proposed instead.

The kernel methods were developed as suitable tools aiming to improve the classification performances of SVM's by increasing the length of the representation of the input data by projecting them into a higher dimensional feature space. The "kernel trick" is the core of a more refined SVM approaches and is essentially based on the particular kernels that offer better generalization without increasing the computational effort. Any kernel method solution comprises two parts: a module that performs the mapping into the embedding or feature space and the learning algorithm designed to discover linear patterns in that space.

The fundamental theoretical result in using symmetric positive definite kernels was established by Mercer (Mercer, 1909). So far several types of kernel methods have been proposed, as for instance SVM's kernel principal component analysis, kernel Gramm-Schmidt, Gaussian processes and base-point machines.

For making SVM more practical, several

algorithms have been developed such as Vapnik's chunking, Osuna's decompositions and Joachims's SVM$^{\text{light}}$. They make the training of SVM possible by breaking the large QP-problem into a series of smaller QP-problems and optimizing only a sub-set of training data patterns at each step. Because the subset of training data patterns optimized at each step is called the working set, these approaches are referred as the working set methods.

Recently, a series of works on developing parallel implementation of training SVM's have been proposed. A parallel SVM is a mixture of SVM's that are trained in simultaneously using sub-sets of the training data set, the results of each SVM being combined by training a multilayer perceptron or by collecting the support vectors in each SVM to train another new SVM.

The content of the survey is structured on five sections. The linear maximal margin classifier in case of linearly separable data and its extension to the nonlinearly separable input data are presented in the first sections of the paper. The fundamentals of SVM kernel-based methods are briefly exposed in the fifth section. The final section is a survey on the main classes of algorithms for solving the dual SVM QP-problems.

2 Linear Maximal Margin Classifier for Linearly Separable Data

SVM learning is among the best "off-the shell" supervised learning algorithms. The task is to predict whether a test sample belongs to one of two classes, on the basis of a finite set of labeled examples

$$\mathcal{S} = \left\{(x_i, y_i), x_i \in \mathbf{R}^n, y_i \in \{-1,1\}, 1 \le i \le N\right\}.$$

The second component of each pair $(x_i, y_i) \in \mathcal{S}$ represents the label of the provenance class of x_i.

The classification (recognition) is performed in terms of a parameterized decision rule $h_{b,w} : \mathbf{R}^n \to \{-1,1\}$,

$$h_{b,w}(x) = g\left(w^T x + b\right),\ g(z) = \begin{cases} 1, & z \ge 0 \\ -1, & z < 0 \end{cases},$$

where $w \in \mathbf{R}^n, b \in \mathbf{R}$.

Note that with respect to the hyperplane of equation $H(w,b): w^T x + b = 0$ the classes labeled 1/-1 correspond to the halfspaces $S^+H(w,b)$ and $S^-H(w,b)$ respectively,

$$S^+H(w,b) = \left\{x \in \mathbf{R}^n / w^T x + b > 0\right\}$$
$$S^-H(w,b) = \left\{x \in \mathbf{R}^n / w^T x + b < 0\right\}.$$

The data $\mathcal{S}$ are called a linearly separable data if there exist w,b such that for each

$$(x_i, y_i) \in \mathcal{S} : \text{if } y_i = 1 \text{ then } x_i \in S^+H(w,b); \text{ if } y_i = -1 \text{ then } x_i \in S^-H(w,b).$$

The property of being linear separable is obviously equivalent to the condition that there exist w,b such that, for each pair (x_i, y_i) of $\mathcal{S}$, $y_i\left(w^T x_i + b\right) > 0$.

For given parameters w,b the functional margin of $H(w,b)$ with respect to the training sample (x_i, y_i) is $\hat{\gamma}_i = y_i\left(w^T x_i + b\right)$.

The correct classification of x_i holds if and only if $\hat{\gamma}_i > 0$, and, from geometric point of view, larger values of the functional margin correspond to more confident predictions. If we define the functional margin of $H(w,b)$ with respect to $\mathcal{S}$ by $\hat{\gamma} = \min\limits_{1 \le i \le N} \hat{\gamma}_i$, then the hyperplane $H(w,b)$ separates without errors $\mathcal{S}$ if and only if $\hat{\gamma} > 0$.

In case when $\mathcal{S}$ is linear separable, the set of parameters (w,b) such that $H(w,b)$ separates without errors $\mathcal{S}$ is infinite and in order to assure confident classifications, the parameters (w,b) that maximize the functional margin $\hat{\gamma}$ should be looked for.

Since, for any $\alpha > 0$,

$S^+H(\alpha w, \alpha b) = S^+H(w,b)$,

$S^-H(\alpha w, \alpha b) = S^-H(w,b)$ and

$H(\alpha w, \alpha b) = H(w,b)$, that is in case both parameters w and b by are multiplied by any positive constant we obtain the same hyperplane and the same decision rule (classifier). Consequently, we can take $\left(\frac{w}{\|w\|},\frac{b}{\|w\|}\right)$ instead of (w,b), that is the vector whose entries are the coefficients of $H(w,b)$ can be assume to be an unit vector orthogonal on $H(w,b)$.

3 Geometric margins

Let $H(w,b)$ be a hyperplane that separates without errors $\mathcal{S}$. For any $(x_i,y_i)\in\mathcal{S}$, let γ_i be the distance of x_i to $H(w,b)$. The value γ_i is the geometric margin of $H(w,b)$ with respect to (x_i,y_i).

Obviously, if $y_i=1$ then

$$\gamma_i=\frac{w^T x_i+b}{\|w\|}=y_i\left(\left(\frac{w}{\|w\|}\right)^T x_i+\frac{b}{\|w\|}\right)$$

and $y_i=-1$, then

$$\gamma_i=\frac{-\left(w^T x_i+b\right)}{\|w\|}=y_i\left(\left(\frac{w}{\|w\|}\right)^T x_i+\frac{b}{\|w\|}\right)$$

that is the values of the geometric margins are not modified if the parameters *w* and *b* are multiply by a positive constant.

The geometric margin of $H(w,b)$ with respect to $\mathcal{S}$ is $\gamma=\min\limits_{1\le i\le N}\gamma_i$.

4 The optimal margin classifier for linearly separable data

For a given training set, a natural desideratum is to find a decision boundary that maximizes the geometric margin, that is to find a classifier that separates the positive and the negative training examples with a "gap" between them (the geometric margin). For a linear separable training set the optimal margin classifier is a hyperplane $H(w,b)$, solution of the constrained optimization problem

$$(1)\begin{cases}\text{maximize }\gamma\\ y_{\mathrm{i}}\left(w^T x_i+b\right)\frac{1}{\|w\|}\ge\gamma, \quad 1\le i\le N\end{cases}$$

that is, we want to maximize γ such that the functional margin of each training sample is at least γ.

Since $\gamma=\frac{\hat{\gamma}}{\|w\|}$, the problem (1) is equivalent to,

$$\begin{cases}\text{maximize }\frac{\hat{\gamma}}{\|w\|}\\ y_{\mathrm{i}}\left(w^T x_i+b\right)\frac{1}{\|w\|}\ge\frac{\hat{\gamma}}{\|w\|}, \quad 1\le i\le N\end{cases}$$

so the problem is equivalent to,

$$\begin{cases}\text{maximize }\frac{\hat{\gamma}}{\|w\|}\\ y_{\mathrm{i}}\left(w^T x_i+b\right)\ge\hat{\gamma}, \quad 1\le i\le N\end{cases}$$

If we impose the condition that the functional margin equals 1, the problem of optimal margin classifier becomes a quadratic programming problem

$$(2)\begin{cases}\text{minimize }\frac{1}{2}\|w\|^2\\ y_{\mathrm{i}}\left(w^T x_i+b\right)\ge 1, \quad 1\le i\le N\end{cases}$$

Any solution of (2) is a canonical optimal margin classifier for the linear separable data $\mathcal{S}$.

In order to solve the problem (2) we use the Lagrange multiplier method. Let $L(w,b,\alpha_1,\alpha_2,\ldots,\alpha_N)$ be the objective function,

$$(3)\quad \begin{aligned}L(w,b,\alpha_1,\alpha_2,\ldots,\alpha_N)&=\frac{1}{2}w^T w\\ &+\sum_{i=1}^{N}\alpha_i\left[1-y_i\left(w^T x_i+b\right)\right]\end{aligned}$$

where $\alpha_1,\alpha_2,\ldots,\alpha_N$ are nonnegative Lagrange multipliers.

The optimal solution is given by the saddle point for L, $\left(w^*,b^*,\alpha^*\right)$, when L is minimized with respect to w and b and maximized with respect to $\alpha_1,\alpha_2,\ldots,\alpha_N\ge 0$. The Karush-

Kuhn-Tucker (KKT) conditions for the objective function (3) are:

$$\left.\frac{\partial L(w,b;\alpha_1,\alpha_2,...,\alpha_N)}{\partial w}\right|_{(w,b;\alpha)=\left(w^*,b^*;\alpha^*\right)}=0$$

$$\left.\frac{\partial L(w,b;\alpha_1,\alpha_2,...,\alpha_N)}{\partial b}\right|_{(w,b;\alpha)=\left(w^*,b^*;\alpha^*\right)}=0$$

$$\alpha_i\left[1-y_i\left(w^{*T}x_i+b^*\right)\right]=0,\quad 1\le i\le N$$

$$\alpha_1,\alpha_2,...,\alpha_N\ \ge 0$$

$$y_i\left(w^{*T}x_i+b^*\right)\ge 1,\quad 1\le i\le N$$

The primal optimization problem:

1. maximize L with respect to α, $\alpha \ge 0$, get $\theta_p(w,b)=\max_{\alpha} L(w,b;\alpha)$
2. minimize $\theta_p(w,b)$ with respect to w,b.

The dual optimization problem:

1. minimize L with respect to w,b, get $\theta_d(\alpha)=\min_{w,b} L(w,b;\alpha)$
2. maximize $\theta_d(\alpha)$ with respect to α, $\alpha \ge 0$.

The optimal solution is $\left(w^*,b^*,\alpha^*\right)$, a saddle point for L, such that

$$L\left(w^*,b^*,\alpha^*\right)=$$
$$\min_{w,b}\max_{\alpha} L(w,b;\alpha)=\max_{\alpha}\min_{w,b} L(w,b;\alpha)$$

The dual problem for L is stated as the unconstrained optimization problem $\underset{w,b}{\text{minimize}}\, L(w,b;\alpha)$

The space of critical points is the set of the solutions of the system

$$\begin{cases}\dfrac{\partial L(w,b;\alpha)}{\partial w}=w-\sum_{i=1}^{N}\alpha_i y_i x_i\ =0\Rightarrow w=\sum_{i=1}^{N}\alpha_i y_i x_i\\[2ex] \dfrac{\partial L(w,b;\alpha)}{\partial b}=-\sum_{i=1}^{N}\alpha_i y_i\ =0\end{cases}$$

Note that the Hessian matrix of L with respect to w, b is, $H_w(L(w,b;\alpha))=I_n$ that is L is minimized in any critical point.

The set of critical points is

$$\left\{(w,b)\Big/\ w=\sum_{i=1}^{N}\alpha_i y_i x_i,\sum_{i=1}^{N}\alpha_i y_i\ =0\right\}.$$

Consequently,

$$\theta_d(\alpha)=\min_{w,b} L(w,b;\alpha)=\frac{1}{2}\left\|\sum_{i=1}^{N}\alpha_i y_i x_i\right\|^2-\sum_{i=1}^{N}\alpha_i\left[y_i\left(\sum_{j=1}^{N}\alpha_j y_j x_j{}^T x_i+b\right)-1\right]=$$
$$=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i{}^T x_j-\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i{}^T x_j-\left(\sum_{i=1}^{N}\alpha_i y_i\right)b+\sum_{i=1}^{N}\alpha_i$$

that is

$$\theta_d(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i{}^T x_j=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\left\|\sum_{i=1}^{N}\alpha_i y_i x_i\right\|^2$$

The maximization of $\theta_d(\alpha)$yields to the constrained concave quadratic problem (QP)

$$(4)\begin{cases}\text{maximize}\,\theta_d(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T x_j\\ \sum_{\mathrm{i}=1}^{\mathrm{N}}\alpha_{\mathrm{i}}y_i=0\\ \alpha_i\geq 0, 1\leq i\leq N\end{cases}$$

Note that the dual Lagrangean $\theta_d(\alpha)$ expressed in terms of the training data depends only on the scalar product of input pattern $\left(x_i^T x_j\right)$, the number of unknown variables being equal to the number of training data.

Let $\alpha^*=\left(\alpha_1^*,\alpha_2^*,...,\alpha_N^*\right)^T$ a solution of the SVM-QP problem (4), consequently the optimal value of the parameter w is $w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i$. The KKT complementarity conditions describing the relationships among the inequality constraints and their associated Lagrange multipliers are,

$\alpha_i^*\left(1-y_i\left(w^{*T}x_i+b\right)\right)=0, 1\leq i\leq N$,

that is, for any $1\leq i\leq N$,

$\alpha_i^*=0$ or $y_{\mathrm{i}}\left(w^{*T}x_i+b\right)=1$.

or, equivalently, if $y_{\mathrm{i}}\left(w^{*T}x_i+b\right)=1$ for any $\alpha_i^*>0$.

The training data x_i for which $\alpha_i^*\neq 0$ are called support vectors. Obviously, for any support vector the relation $y_{\mathrm{i}}\left(w^{*T}x_i+b\right)=1$ holds, that is x_i belongs to the hyperplane $H\left(w^*,b\right)$ and $\alpha_i^*\neq 0$ is called active Lagrange multipliers.

Obviously, if α_i^* is an active multiplier, then $y_{\mathrm{i}}\left(w^{*T}x_i+b\right)-1=0$, that is $w^{*T}x_i+b=y_i$, or equivalently, $\hat{\gamma}_i=\hat{\gamma}$.

The parameter b can not be explicitly computed by solving the SVM problem. A suitable value $b*$ of b such that $y_i\left(w^{*T}x_i+b^*\right)\geq 1$ holds for all input data.

If $y_i=1$, then

$\left(w^{*T}x_i+b^*\right)\geq 1\Leftrightarrow$

$b^*\geq 1-w^{*T}x_i=y_i-w^{*T}x_i$, that is $b^*\geq\max\limits_{\substack{i\\ y_i=1}}\left(y_i-w^{*T}x_i\right)=1-\min\limits_{\substack{i\\ y_i=1}}w^{*T}x_i$

If $y_i=-1$, then $\left(w^{*T}x_i+b^*\right)\leq -1\Leftrightarrow b^*\leq -1-w^{*T}x_i$, that is $b^*\leq\min\limits_{\substack{i\\ y_i=-1}}\left(-1-w^{*T}x_i\right)=-1-\max\limits_{\substack{i\\ y_i=-1}}w^{*T}x_i$

Consequently, a suitable value of b^* should be selected such that

$1-\min\limits_{\substack{i\\ y_i=1}}w^{*T}x_i\leq b^*\leq -1-\max\limits_{\substack{i\\ y_i=-1}}w^{*T}x_i$

In case the middle of the interval $\left[1-\min\limits_{\substack{i\\ y_i=1}}w^{*T}x_i, -1-\max\limits_{\substack{i\\ y_i=-1}}w^{*T}x_i\right]$ is selected, we get

$b^*=-\frac{1}{2}\left\{\max\limits_{\substack{i\\ y_i=-1}}w^{*T}x_i+\min\limits_{\substack{i\\ y_i=1}}w^{*T}x_i\right\}$.

The resulted classifier corresponds to the decision rule

$D(x)>0\Rightarrow x$ is classified in h_1

$D(x)<0\Rightarrow x$ is classified in h_2

$D(x)=0 \Rightarrow x$ is unclassifiable,
where $D(x)=w^{*T}x+b^{*}$.

5 Nonlinear SVM's classifier

In general, classes are not only overlapped, but the genuine separation functions are nonlinear hypersurfaces. Consequently, a more general type of SV machines is required to create nonlinear decision hypersurfaces, and able to classify nonlinearly separable data. This can be achieved by considering the linear classifier in the so-called feature space of higher dimension than the dimension of the initial input space.

The basic idea of designing nonlinear SVM's is to consider $g:\mathbf{R}^n \rightarrow \mathbf{R}^s$ that maps the input space $\mathbf{R}^n$ onto the feature space $\mathbf{R}^s$, where $s>n$. Consequently, to each input vector $x_i \in \mathbf{R}^n$ corresponds the *s*-dimensional representation $g(x_i)=(g_1(x_i),g_2(x_i),...,g_s(x_i))$. The SVM-QP problem (4) in the feature space becomes,

$$(5)\begin{cases}\text{minimize}-\left(\sum_{i=1}^{N}\alpha_i+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (g(x_i))^T g(x_j)\right)\\ \sum_{\mathrm{i=1}}^{\mathrm{N}}\alpha_{\mathrm{i}}y_i=0\\ \alpha_i\geq 0, 1\leq i\leq N\end{cases}$$

There are two main problems when performing the mapping g on one hand the choice of g should result in a reach class of decision hyperplanes and, on the other hand, the computation of the scalar product $(g(x))^T g(x)$could become very challenging if the dimensionality of the feature space is very large. This explosion in dimensionality can be avoided by noticing that in the QP-problem (5) as well as in the final expression for classifier the training data only appear in the form of the scalar product $(g(x_i))^T g(x_j)$, that is the solution can be expressed in terms of the kernel function $k(x,x')=(g(x))^T g(x')$. Since the kernel function $k(x,x')$is a function defined on the input space, using a kernel function allows to avoid performing a mapping $g(x)$at all. In this way, a possibly extremely high dimensionality of the feature space can be bypassed and, depending on the particular kernel, SVM's operating in an infinite dimensional space can be designed. Moreover, by applying kernels, we do not have to know what the actual mapping $g(x)$is.

There is a large class of possible kernels because a kernel should only fulfill the Mercer's conditions (Mercer, 1909). In order to avoid the explosion of computational complexity, the choice of the kernel should be such that the computation of $k(x_i,x_j)=(g(x_i))^T g(x_j)$is in fact carried out in the initial input space, this property being known under the name of "the kernel trick". For instance, in case $n=2$, if we consider the maps g_1,g_2 : $\mathbf{R}^2 \rightarrow \mathbf{R}^3$, g_3 : $\mathbf{R}^2 \rightarrow \mathbf{R}^4$, and g_4 : $\mathbf{R}^2 \rightarrow \mathbf{R}^6$ defined by,

$$x=(x_1,x_2)^T$$
$$g_1(x)=\left(x_1^2,\sqrt{2}x_1x_2,x_1^2\right)$$
$$g_2(x)=\left(x_1^2-x_2^2,2x_1x_2,x_1^2+x_2^2\right)$$
$$g_3(x)=\left(x_1^2,x_1x_2,x_1x_2,x_2^2\right)$$
$$g_4(x)=\left(1,\sqrt{2}x_1,\sqrt{2}x_2,\sqrt{2}x_1x_2,x_1{}^2,x_2^2\right)$$

and we denote by $k_i(x,x')=(g_i(x))^T g_i(x'), i=1,2,3,4$, we get

$$k_i(x,x')=\left(x^T x'\right)^2, i=1,2,3$$

$k_4(x,x')=\left(1+x^T x'\right)^2$

To each of the maps $g_i, i=1,2,3,4$ corresponds a different nonlinear decision hypersurface in the input space, but the computations in a higher dimension feature space can be carried out in the input space.

Some of the most frequently used positive definite kernels are

- $k(x,x')=x'^T\ x$ - linear, dot product kernel;
- $k(x,x')=\left(x'^T\ x+1\right)^d$ - complete polinomial of degree *d*;
- $k(x,x')=\exp\left\{-\frac{\left\|x-x'\right\|_2^2}{2\sigma^2}\right\}$ - Gaussian RBF;
- $k(x,x')=\tanh\left(\kappa x'^T\ x+\theta\right)$ - multilayer perceptron;
- $k(x,x')=\frac{1}{\sqrt{\left\|x-x'\right\|^2+\beta}}$ - inverse multiqadric kernel.

In case *k* is a positive definite kernel, the Gramm matrix $G=\left\|k\left(x_i,x_j\right)\right\|$ is a positive definite matrix, and moreover any symmetric positive definite matrix can be regarded as a kernel matrix, that is an inner product matrix in some space.

In case a certain positive definite kernel *k* that fulfills Mercer's conditions, the SVM QP-problem (4) becomes,

$$(6)\begin{cases}\text{minimize}-\left(\sum_{i=1}^{N}\alpha_i+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k\left(x_i,x_j\right)\right)\\ \sum_{\text{i}=1}^{\text{N}}\alpha_{\text{i}} y_i=0\\ \alpha_i\geq 0, 1\leq i\leq N\end{cases}$$

In the more general case when the input data set $\mathcal{S}$ is nonlinear separable, the set of feasible solutions is empty. A natural extension of the SVM in this case is represented by modifying the objective function of the problem (2) to include the effect of misclassifications. In this case, the original SVM problem is formulated as

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

where $c>0,\sigma\geq 1$ are convenient selected constants and *F* is a convex function such that $F(0)=0$, and $\xi_1,\xi_2,\ldots,\xi_N$ are the slack variables. The meaning of the slack variables is that the example x_i for which $\xi_i\geq 1$ is misclassified by the hyperplane $H(w,b): w^T x+b=0$.

In the particular case when $\sigma=1$ and $F(u)=u$, using the Lagrange multipliers method, we get the objective functions,

$$Q(w,b,\xi,\alpha,\beta)=\frac{1}{2}\left\|w\right\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\alpha_i\left(y_i\left(w^T x_i+b\right)-1+\xi_i\right)-\sum_{i=}^{N}\beta_i\xi_i$$

where $\alpha = (\alpha_1, ..., \alpha_N)^T$ and $\beta = (\beta_1, ..., \beta_N)^T$ are nonnegative Lagrange multipliers. Using the KKT conditions,

$$\frac{\partial Q(w,b,\xi,\alpha,\beta)}{\partial w} = 2w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0$$

$$\frac{\partial Q(w,b,\xi,\alpha,\beta)}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\frac{\partial Q(w,b,\xi,\alpha,\beta)}{\partial \xi} = c * \mathbf{1} - \alpha - \beta = 0$$

$$\alpha_i \left(y_i \left(w^T x_i + b \right) - 1 + \xi_i \right) = 0, \quad 1 \le i \le N$$

$$\beta_i \xi_i = 0, \quad 1 \le i \le N$$

$$\alpha_i \ge 0, \quad \beta_i \ge 0, \quad \xi_i \ge 0, \quad 1 \le i \le N$$

we get

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\alpha_i + \beta_i = C, \; 1 \le i \le N$$

Consequently, the dual problem becomes the constrained QP-problem

$$(8)\begin{cases} \text{minimize}\left(-\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j x_i^T x_j \right) \\ \sum_{i=1}^{N} y_i \alpha_i = 0, \quad C \ge \alpha_i \ge 0, \quad 1 \le i \le N \end{cases}$$

The parameter w^* of the soft margin (L1-SVM) hyperplane is

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$$

where $\alpha^* = \left(\alpha_1^*, \alpha_2^*, ..., \alpha_N^* \right)$ is a solution of (8).

1. If $\alpha_i^* = 0$, since $\alpha_i^* + \beta_i = C$ and $\beta_i \xi_i = 0$, we get $\beta_i = C \ne 0,$ that is $\xi_i = 0$ which means that the example x_i is correctly classified.

2. If $0 < \alpha_i^* < C$ then $\left(y_i \left(w^T x_i + b \right) - 1 + \xi_i \right) = 0$, since $\alpha_i^* + \beta_i = C$ and $\beta_i \xi_i = 0$, we get $\beta_i \ne 0,$ that is $\xi_i = 0$ and $y_i \left(w^T x_i + b \right) = 1$. In this case x_i is correctly classified. The example x_i is called an unbounded support vector.

3. If $\alpha_i = C$ then $\beta_i = 0$ and $\xi_i \ge 0$, that is $y_i \left(w^T x_i + b \right) = 1 - \xi_i$. We say that x_i is a bounded support vector, which is correctly classified if $0 \le \xi_i < 1$ and misclassified if $\xi_i \ge 1$ respectively.

If we denote by S a set of support vectors and by U the set of unbounded support vectors, the expression of the decision function is,

$$D(x) = \sum_{x_i \in S} \alpha_i y_i x_i^T x + b$$

where $b = \frac{1}{|U|} \sum_{x_i \in U} \left(y_i - w^T x_i \right)$.

The classification decision is

$$\begin{cases} D(x) > 0 \Rightarrow x \in h_1 \\ D(x) < 0 \Rightarrow x \in h_2 \end{cases}$$

Note that in case U=S, the set $D(x) = 0$ is the generalization region.

A natural extension to the case when the input data are projected in a higher dimension feature space by the map g:$\mathbf{R}^n \rightarrow \mathbf{R}^s$, $s > n$, yields to the objective function,

$$Q(\alpha) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j g(x_i)^T g(x_j)$$

or, using the kernel $k(x, x')$,

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

In this case, the dual SVM QP-problem becomes,

$$(9)\begin{cases} \text{minimize}\left(-\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j k(x_i, x_j) \right) \\ \sum_{i=1}^{N} y_i \alpha_i = 0, \quad C \ge \alpha_i \ge 0, \quad 1 \le i \le N \end{cases}$$

Again, the only difference to the separable nonlinear classifier is the upper bound C on the Lagrange multipliers α_i. In this way, we limit the influence of training data points that will remain on the “wrong” side of a separating nonlinear hypersurface. After the dual variables are computed, the decision hypersurface $D(x)$ is

$$D(x) = \sum_{i=1}^{N} \alpha_i^* y_i k(x, x_i) + b = \sum_{x_i \in S} \alpha_i^* y_i k(x, x_i) + b$$

where S a set a support vectors.

The existence and calculation of the bias *b* is now not a direct procedure as it is for a linear hyperplane, depending upon the particular kernel the bias *b* can be implicitly part of the kernel function. A more detailed discussion and analysis can be found in (Kecman, Huang, Vogt, 2005) as well as in (Vogt, Kecman, 2005).

6 Methods to solve the dual SVM QP-problems

Methods to solve the corresponding dual QP optimization problems to SVM learning include Sequential Minimal Optimization (SMO), decomposition methods (Smola, Schőlkopf, 1999) and (Laskov, 2002), and methods to solve the least squares SVM formulations (Cawley, Talbot, 2002), (Keerthi, Shevade, 2003) and (Suykens, De Brabanter, Lukas, 2002) as well as software packages as SVM^{light} (Joachims,1998.), mysvm (Rueping, 2003) and many others.

Since the size of the QP problems depends on the number of data, the problem can not be straightforward solved via standard QP techniques.

In 1982 Vapnik (Vapnik, 1982) proposed a method to solve the QP problem arising from SVM referred as “chunking”. The idea of the chunking algorithm uses the fact that the value of the quadratic form remains unchanged if the rows and columns of the matrix of entries $\left\| y_i y_j x_i^T x_j \right\|$ corresponding to zero Lagrange multipliers are removed. Therefore, the large QP problem can be split into smaller QP problems whose ultimate goal is to identify all the non-zero Lagrange multipliers and discard all the zero ones. Chunking reduces significantly the size of the matrix corresponding to the particular QP problem, but still can not handle any large scale training problem.

A new class of QP algorithms for SVM derived from the developments proposed by Osuna (Osuna, Freund and Girosi, 1997). Briefly, Osuna proved that the large QP problem can be reduced to a series of smaller QP sub-problems based on the idea that as long as at least one example that violets the KKT conditions is added to the examples used in the previous sub-problem, at each step the overall objective function is reduced and a feasible point that obeys the constraints is maintained. This way the algorithm proposed by Osuna performs by adding one example and subtracting another example at each step.

Sequential minimal optimization (SMO) algorithm proposed by Platt (Platt, 1998) 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 using Osuna’s theorem (Osuna, Freund and Girosi, 1997). Unlike the previous methods, the SMO algorithm solves the smallest optimization problem at each step. Obviously, in case of the standard SVM-QP problem the smallest possible optimization problem involves two Lagrange multipliers, because the Lagrange multipliers must fulfill a linear inequality constraint. The SMO algorithm performs by choosing two Lagrange multipliers to jointly optimize, finds the optimal values for these multipliers and updates the SVM accordingly.

The convergence of the SMO algorithm is guaranteed by the Osuna’s theorem because one of the Lagrange multipliers selected at each step violates the KKT conditions, that is at each step the value of the objective function decreases. In order to speed convergence, several heuristics for selecting the two Lagrange multipliers have been proposed.

For the various test sets, the training time required by SMO empirically scales between N and $N^{2.2}$. The training time of chunking scales between $N^{1.2}$ and $N^{3.4}$. The scaling of SMO can be more than one order better than chunking. For the real world test sets, SMO can be a factor of 1200 times faster for linear SVM's and a factor of 15 times for nonlinear SVM's. Because of its ease of use and better scaling with training set size, SMO is a strong candidate for becoming the standard SVM training algorithm.

Generalizations and improvements have been recently proposed by many others. For instance, in (Cao, Keerthi, Ong & all, 2006) a parallel version of SMO is proposed for fast training SVM. Unlike the sequential SMO algorithms, which handle all the training data points using one CPU processor, the parallel SMO first partitions the entire training data set into smaller subsets and then simultaneously runs multiple CPU processors to deal with each of the partitions data sets.

Also, in order to improve the generalization capacities, several approaches have been communicated. For instance, in (Sanchez, 2003) it is proposed a method to improve the generalization capabilities of SVM classifiers based on the enhancement of the special resolution on the boundary surface by introducing a conformal mapping into the Riemannian geometry induced by the classifier kernel function. Several experimental results pointed out the validity of these information-geometrical considerations as an approach to optimal, data-dependent SVM kernel choice and generalization improvement.

Another example of an approach for improving generalization when SVM learning is applied to RBF networks determines the optimal spread parameter for a Gaussian kernel in classification and regression problems using the Fisher discrimination and scale space theory respectively.

References

[1] K. Bennett, "Combining Support Vector and Mathematical Programming Methods for Classification," *Advances in Kernel Methods – Support Vector Learning*, Cambridge, MA, MIT Press, 1999.

[2] G.C. Cawley and N.L.C. Talbot, "Improved sparse least squares support vector machines," *Neurocomputing* 48 (1-4), 2002.

[3] L.J. Cao, S.S. Keerthi, C.J. Ong, P. Uvaraj, X.J. Fu and H.P. Lee, "Developing parallel sequential minimal optimization for fast training support vector machine," *Neurocomputing*, vol. 70, pp.93-104, 2006.

[4] S.S. Keerthi and S.K. Shevade, "SMO algorithm for least squares SVM formulations," *Neural Compt*, vol. 15, no. 2, 2003.

[5] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proceedings of the European Conference on Machine Learning (ECML)*, Springer, 1998.

[6] T. Joachims, "Making Large-Scale SVM Learning Practical," in *Schőlkopf, B., Burges, C.J., Smola, A.J. eds. Advances in Kernel Methods – Support Vector Learning*, Cambridge, MA, MIT Press., 1998.

[7] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines," *Proceedings of the International Conference on Machine Learning (ICML)*, 1999.

[8] T. Joachims, "A Statistical Learning Model of Text Classification with Support Vector Machines," *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR), ACM*, 2001.

[9] V. Kecman, T.M. Huang and M. Vogt, "Iterative Single Data Algorithm for Training Kernel Machines From Huge Data Sets: Theory and Performance," *Studies in Fuzziness and Soft Computing*, vol. 177, pp 255–274. Springer Verlag, 2005.

[10] P. Laskov, "An improved decomposition algorithm for regression support vector machines," *Mach.*

Learning, vol. 46, 2002.

[11] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Philosophical Transactions of the Royal Society of London*, Series A 209, 1909.

[12] E. Osuna, R. Freund and F. Girosi, "An improved training algorithm for support vector machines," *Neural Networks for Signal Processing VII – Proceedings of the 1997 IEEE Workshop*, New York, 1997.

[13] S. Rueping, "mySVM: another one of those support vector machines," Available: http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM, 2003.

[14] V.D.A. Sanchez, "Advanced support vector machines and kernel methods," *Neurocomputing, vol.* 55 (2003), pp.5-20, 2003.

[15] B. Schőlkopf, C.J. Burges and A.J. Smola, *Advances in Kernel Methods – Support Vector Learning*, Cambridge, MA, MIT Press. 1999.

[16] B. Schőlkopf, K. Sung, C.J. Burges, F. Girosi, P. Nyiogi, T. Poggio and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. Sign. Proc.*, vol. 45, 1997.

[17] B. Schőlkopf, P.Y. Simard, A.J. Smola and V. Vapnik, "Prior knowledge in support vector knowledge," *Advances in Kernel Methods – Support Vector Learning*, Cambridge, MA, MIT Press, 1998.

[18] A.J. Smola, B. Schőlkopf and K.R. Müller, "General Cost Functions for Support Vector Regression," *Proc. of the Ninth Australian Conf. n Neural Networks*, Brisbane, Australia, 1998.

[19] A.J. Smola and B. Schőlkopf, "A Tutorial on Support Vector Regression," *NeuroCOLT2 Technical Report Series*, 1999.

[20] M. Stitson and all, "Support Vector Regression with ANOVA Decomposition Kernels," *Advances in Kernel Methods – Support Vector Learning*, Cambridge, MA, MIT Press, 1999.

[21] J.A.K. Suykens, L. Lukas and J. Vanderwalle, "Sparse approximation using least squares support vector machines," *IEEE International Symposium on Circuits and Systems, ISCAS 2000*, Geneva, Switzerland, May 2000.

[22] J.A.K. Suykens, J. Vanderwalle and B. DeMoor, "Optimal control by least squares support vector machines," *Neural Networks*, Vol. 14, 2001.

[23] J.A.K. Suykens, J. De Brabanter and L. Lukas, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing special issue*, 2002.

[24] J.A.K. Suykens, "Support vector machines: a nonlinear modeling and control perspective," *Technical Report*, April 2001.

[25] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, N.Y., 1995.

[26] V. Vapnik and Chervonenkis, "A note of one class of perceptrons," *Automation and Remote Control*, vol. 25, 1964.

[27] V. Vapnik, S. Golowich and A. Smola, "Support vector method for functions approximation, regression estimation, and signal processing," *Advances in Neural Information Processing Systems 9*, Cambridge, MA, MIT Press. 1997.

[28] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer Verlag, Berlin, 1982.

[29] V. Vapnik, *Statistical Learning Theory*. John Wiley, N.Y., 1998.

[30] V. Vapnik, "The support vector method of function estimation," *Nonlinear Modeling: Advanced Black-box Technique*, Kluwer Academic Publishers, Boston, 1998.

[31] M. Vogt and V. Kecman, "Active-Set Method for Support Vector Machines," *Studies in Fuzziness and Soft Computing*, vol.177, pp 133–178. Springer Verlag, 2005.

[32] G. Wahba, Y. Lin and H. Zang, "GACV for support vector machines," *Advances in Large Margin Classifiers*,

MIT Press, Cambridge MA, 2000.

[33] Y. Yue, T. Finley, F. Radlinski and T. Joachims, “A Support Vector Method for Optimizing Average Precision,” *Proceedings of the Conference on Research and Development in Information Retrieval* (SIGIR), 2007

Luminita STATE, Professor, PhD, currently working with University of Pitesti, Department of Mathematics and Computer Science. Competence areas: artificial intelligence, machine learning, statistical pattern recognition, digital image processing. Research in the fields of machine learning, pattern recognition, neural computation. Author of 15 books and more than 120 papers published in national and international journals.

Catalina-Lucia COCIANU, Professor, PhD, currently working with Academy of Economic Studies, Faculty of Cybernetics, Statistics and Informatics, Department of Informatics in Economy. Competence areas: statistical pattern recognition, digital image processing, machine learning. Research in the fields of pattern recognition, data mining, signal processing. Author of 12 books and more than 80 papers published in national and international journals.

Doina FUSARU, Professor, PhD, currently working with Spiru Haret University, MFC, Department of Accounting and Management Information System. Competence areas: auditing, data base systems, information systems, pattern recognition, programming languages. Research in the fields of auditing, data bases, information systems, pattern recognition. Author of 14 books and more than 40 papers published in national and international journals.