

A Novel Vectors Surgeon Machine Based on Statistical Learning Theory

Yu LIANG and Liang-zhi GAN*

School of Electrical Engineering and Automation, Jiangsu Normal University, 221116, China

*Corresponding author

Keywords: Support vector machine, Optimal brain surgeon, Statistical learning theory.

Abstract. Support vector machine (SVM) was first used to solve classification problems and then developed to solve regression problems. Despite its widespread success, the SVM suffers from some important limitations, one of the most significant being that its optimization algorithms to train the SVM are complicated. We investigate the problem of training SVM and present a novel machine learning method that is called Vectors Surgeon Regression Machine (VSRM). The VSMR minimizes the squares error and controls the structural risk by the number of parameters, which is equal to the VC dimension of the set of functions. The optimal brain surgeon usually used in neural networks is introduced to prune the support vectors. The proposed method is tested on classification data sets.

Introduction

Support Vector Machine (SVM) is based on the statistical learning theory\cite{1}. Structural Risk Minimization (SRM) principle has been successfully used on SVM and many people are abstracted by its excellent performance. But Training a support vector machine (SVM) is difficult because no high quality QP solvers are available. Based on standard SVM, a lot of variations of SVM have been put forward such as Least Squares Support Vector Machine (LS-SVM)\cite{2}.

In a regression problem, one is supplied with a set of data points $D = \{(x_i, y_i), \cdots, (x_N, y_N)\} \subset \mathbb{R}^d \times \mathbb{R}$ that are sampled from an unknown function, where $x_i, i = 1, \cdots, N$, are input points and $y_i, i = 1, \cdots, N$, output points. The goal is to find a fit to the data points such that an approximation to the unknown function is obtained. A support vector machine is an approximation of the form:

$$f(x, w) = \sum_{i=1}^{N} w_i \varphi_i(x) + b,$$

where the functions $\{\varphi_i(x)\}_{i=1}^{M}$ are features and $w = \{w_1, \cdots, w_M\}$ are the $M$ parameters of the model. The model parameters are chosen to minimize the functional:

$$\min_w w^T w + C \sum_{i=1}^{M} (\xi_i^+ + \xi_i^-)$$

subject to

$$\begin{align*}
\xi_i^+, \xi_i^- \geq 0, \\
y_i - f(x_i, w) - b &\leq \epsilon + \xi_i^+, \\
f(x_i, w) + b - y_i &\leq \epsilon + \xi_i^-.
\end{align*}$$

This optimization formulation can be transformed into the dual problem, and its solution is given by

$$f(x) = \sum_{i=1}^{M} (\alpha_i - \alpha_i^+) \varphi(x) \varphi(x) + b.$$
where $\alpha_i, \alpha_i^*$ and $b$ are the solutions to QP (2). If there were a “kernel function” $k(x_i, x)$ satisfying $k(x_i, x) = \varphi(x_i)^T \varphi(x)$, we would only need to use $k(x_i, x)$ in the training algorithm, and would never need to explicitly even know what $\varphi(x)$ is \cite{3,4}. One example is:

$$k(x_i, x) = \varphi(x_i)^T \varphi(x) = e^{-\frac{||x_i - x||^2}{\sigma^2}} \quad (4)$$

In this example, $\varphi(x)$ is infinite dimensional \cite{5,6}, and we have defined dot product by (4). The solution of problem (2) could be written as equation (1), where $w_i = \alpha_i - \alpha_i^*$, $\varphi_i(x) = k(x_i, x) = e^{-\frac{||x_i - x||^2}{\sigma^2}}$.

**Analysis of SVMs**

The SVM suffers from some important limitations, one of the most significant being that its optimization algorithms to train the SVM are complicated. Training SVM is done by mapping the underlying optimization problem into a quadratic programming (QP) problem. Unfortunately, high quality QP solvers are not readily available. The second limitation is

Some important conclusion has been drawn by Vapnik \cite{1} and his cooperators:

**Lemma 1** The VC dimension of a set of functions that are linear in the parameters

$$f(z, a) = \sum_{i=1}^{2^h} \alpha \varphi(z) + \alpha_0$$

equals $n$, the numbers of parameters of a set of functions.

**Lemma 2** With probability $1 - \eta$ simultaneously for all functions in a set of real-valued bounded functions $Q(z, w), w \in A$, the inequality

$$R(w) \leq R_{emp}(w) + (B - A) \sqrt{\varepsilon(l)}$$

is valid, where

$$\varepsilon(l) = \frac{H^{\lambda, B}(2l) - \ln \eta / 4}{l} + \frac{1}{l}$$

where $R(w)$ is the expected risk, $R_{emp}(w)$ is the empirical risk, and, $H^{\lambda, B}(2l)$, the annealed entropy.

**Lemma 3** The growth function $G^\lambda(l)$ of a set of functions either (a) satisfying the equality

$$G^\lambda(l) = l \ln 2$$
or (b) is bounded by the inequality

$$G^\lambda(l) \leq h(\ln(l / h) + 1)$$

where $h$ is the VC dimension.

**Lemma 4** The following inequality holds true:

$$H^{\lambda, B}_{ann}(l) \leq G^\lambda(l) \leq h(\ln(l / h) + 1)$$

where $h$ is the VC dimension, $G^\lambda(l)$ is the growth function of a set of functions, and $H^{\lambda, B}_{ann}(l)$ is the annealed entropy.

From lemma 2 to lemma 4 we can draw a conclusion:
\[ R(w) \leq R_{emp}(w) + (B - A) \sqrt{\frac{h \ln(2l/h) - \ln \eta / 4 + 1}{l}}. \]  

(5)

And from lemma 1, we can get the VC dimension \( h \) of the set of the functions. So inequality (5) could be used to find the minimum of the expected risk.

**Theorem 1** The number of support vectors of SLS-SVM is finite if all the samples come from a limited area.

Proof. Firstly we proof that the distance between any of the two support vectors in \( \mathbb{R}^d \) is bigger than a constant \( \delta \). According to the Support Vectors Selecting Method introduced in section 3, we have:

\[
(\phi(x_h) - \phi(x_l))^T (\phi(x_h) - \phi(x_l)) \\
\geq \min (\phi(x_h) - \sum_{i=1}^{h-1} \lambda_i \phi(x_i))^T (\phi(x_h) - \sum_{i=1}^{h-1} \lambda_i \phi(x_i)) \\
\geq \varepsilon.
\]

This means that the distance between last support vector \( \phi(x_h) \) and any other support vector \( \phi(x_i) \) \( (i = 1, \ldots, h - 1) \) is bigger than \( \varepsilon \). For the same reason, the distance between the j-th support vector \( \phi(x_j) \) and any other support vector \( \phi(x_i) \) \( (i = 1, \ldots, j - 1) \) is bigger than \( \varepsilon \). So the following inequality holds true to any \( \phi(x_i) \) and \( \phi(x_j) \) which are belongs to \( X_h \):

\[
(\phi(x_j) - \phi(x_i))^T (\phi(x_j) - \phi(x_i)) \\
= 2 - 2\varepsilon \frac{1}{1 - 1/\sigma^2} \geq \varepsilon.
\]

We can easily know that \( \|x_j - x_i\|^2 \geq \delta(\varepsilon, \sigma) \). All the samples come from a limited area, so the number of support vectors in \( X_h \) is finite.

**Vectors Surgeon Regression Machine**

Hassibi provided the method that is called optimal brain surgeon to improve generalization of networks. This method could be used to control the VC dimension directly.

Supplied with a set of data points \( D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \) that are sampled from an unknown function, the following method is used to find the regression function:

minimize \( E = e^T e \)

subject to

\[ y_j = \sum_{i=1}^{N} \omega_i \phi(x_j) + b + e_j, \]

where \( e = [e_1; \ldots; e_N], (x, y), i = 1, \ldots, N \) are sample points. Different from QP (2), the solution of problem (6) is simple because its constraint conditions are equalities. Let \( Y = [y_1; \ldots; y_N] \), \( I = [1; \ldots; 1_N] \), \( \theta = [b; w_1; \ldots; w_M] \), \( (K)_{ij} = \phi_j(x_i) \), \( \Phi = [I; K] \), the empirical risk can be represented as:

\[ E = (Y - \Phi \theta)^T (Y - \Phi \theta) \]

(7)

The vector \( \theta \) can be represented as:
\( \theta = (\Phi^T \Phi)^{-1} \Phi^T Y \). \hfill (8)

The method only minimizes the empirical risk. To control the structural risk, the following optimal brain surgeon method introduced by Hassibi is used to control the VC dimension.

The functional Taylor series of the error with respect to parameters is \( E \) \( \frac{\partial E}{\partial \theta} = \theta \Phi \Phi \Phi Y \). \hfill (8)

From (7), we know that \( \frac{\partial^2 E}{\partial \theta^2} = 0 \). The empirical risk \( E \) is minimized, so \( \frac{\partial E}{\partial \theta} = 0 \). If the estimated expected risk \( R(w) \) decreases, then the \( q \)th vector should be deleted, and we proceed to step 4; otherwise go to step 5. (Other stopping criteria can be used too.)

The \( L_q \) is called “saliency” of vector \( q \), the increase in error that results when the vector is eliminated. Now reviewing Eq. (1) and lemma 1, we know that the VC dimension \( h \) of the set of functions is \( M + 1 \), the number of parameters. So we find a new way to control the structural risk: vector surgeon \( [9,10] \). Thus we have the algorithm of Table 1.

<table>
<thead>
<tr>
<th>Table 1. Vectors Surgeon Regression Machine procedure.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Train a Vectors Surgeon Regression Machine with all sample points to minimize error.</td>
</tr>
<tr>
<td>2) Computing ( H^{-1} ).</td>
</tr>
<tr>
<td>3) Find the ( q ) that gives the smallest saliency ( L_q = \frac{1}{2} \left[ H^{-1} \right]_{qq} ). If the estimated expected risk ( R(w) ) decreases, then the ( q )th vector should be deleted, and we proceed to step 4; otherwise go to step 5. (Other stopping criteria can be used too.)</td>
</tr>
<tr>
<td>4) Use the ( q ) from step 3 to update all parameters (Eq. 11). Go back to step 2.</td>
</tr>
<tr>
<td>5) No more vectors can be deleted, training procedure ends.</td>
</tr>
</tbody>
</table>

The difficulty appears to be in step 2 in the VSRM procedure, since inverting a matrix of thousands or millions of terms seems computationally expensive, or a matrix may be singular or bad conditioned. In what follows we shall give some skills to overcome this difficulty.

We could find that \( w \) is the linear combination of all samples in feature space. By solving the following minimizing problem, we know whether \( \phi(x_i) \) is one of the basis or not.

\[
\min \ f(\lambda) = (\phi(x_i) - \sum_{i,k} \lambda_i \phi(x_i))^T (\phi(x_i) - \sum_{i,k} \lambda_i \phi(x_i)) \hfill (13)
\]

If \( f(\lambda) = 0 \), \( \phi(x_i) \) is not one element of the basis because it can be linearly represented by other elements. If \( f(\lambda) > 0 \), \( \phi(x_i) \) is one of the basis. According to Kuhn-Tucker conditions, we have:
where \( K_0 = (k(x_1, x_1), \ldots, k(x_{N-1}, x_{N-1}))^T \), \( K \) is a square matrix, \( K_g = k(x_i, x_j) \), Mercer condition has been applied to matrix \( K \), so \( K \) is positive defined and \( K^{-1} \) exists. We have \( \lambda_{\text{min}} = K^{-1}K_0 \).

The set \( X = \{x_1, \ldots, x_N\} \) is denoted as \( X' \) in feature space, that is \( X' = [\varphi(x_1), \ldots, \varphi(x_N)] \). We can split \( X' \) into two parts: \( x_h \) and \( x_A \). All the elements in \( x_h \) are linearly independent and the elements in \( x_A \) can be linearly approximated by \( x_h \). The elements in set \( x_h \) are different from each other, so we can look it as a row vector: \( x_h = [\varphi(x_1), \ldots, \varphi(x_N)] \).

Since we have known that \( w = \sum_{k=1}^N \alpha_k \varphi(x_k) \) and the elements in \( x_A \) can be approximately represented by \( x_h \), so \( w \) could be write as \( w \approx \sum_{\varphi(x_i) \in x_h} \theta \varphi(x_i) = X_h \theta^T \), rewrite formula (1) as:

\[
\min_{\theta, b} \Phi(\theta, b, e) = \frac{1}{2} (X_h \theta)^T (X_h \theta) + \frac{1}{2} \gamma \sum_{k=1}^N e_k^2
\]

subject to \( y_k [(X_h \theta)^T \varphi(x_k)] + b = 1 - e_k, k = 1, \ldots, N \)

where \( \theta = [\theta_1, \ldots, \theta_h] \), \( h \) is the number of elements in set \( x_h \).

Dimension of equations set (8) is \( h + 1 \) \([11, 12]\). It is completely controlled by the number of elements in set \( x_h \). According to Christopher J.C. Burges, if the kernel function is \( k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / \sigma^2) \), the width parameter \( \sigma \) can control the VC dimension of RBF SVM. When \( \sigma \rightarrow 0^+ \), the VC dimension of RBF SVM inclines to infinite; or when \( \sigma \rightarrow +\infty \), it inclines to zero. So \( \sigma \) is an important parameter. A suitable \( \sigma \) can control both the learning ability of SL-SVM and the dimension of vector \( x_h \). Example in section 5 shows that bigger the width \( \sigma \) is, smaller the dimension of vector \( x_h \) is, just as pointed out by Christopher J.C. Burges \([13,14]\).

The second skill is called the **Incremental and Decrement Inverting Method**. This method gives the way to inverting a matrix of thousands or millions of terms.

**Lemma 5** For matrix \( A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \), where \( A_{11}^{-1}, A_{22}^{-1} \) exist, the following equation is true

\[
A^{-1} = \begin{bmatrix} A_{11}^{-1} & A_{12}^{-1} \left[ A_{21} A_{11}^{-1} A_{22}^{-1} - A_{22}^{-1} \right] \\ A_{21} A_{11}^{-1} - A_{22}^{-1} & A_{22}^{-1} \end{bmatrix}
\]

**Lemma 6** For matrix \( A, B, C, D \), where \( A^{-1}, C^{-1} \) exist, the following equation is true

\[(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}\]

According to reference [1], we have the following conclusions.

**Theorem 2** The SLS-SVM is consistency of ERM for any probability measure.

Proof. According to lemma 1, the VC dimension of the SLS-SVM is \( h + 1 \), where \( h \) is the number of support vectors according to lemma 1. So

\[
0 \leq \lim_{l \to \infty} \frac{G^A(l)}{l} \leq \lim_{l \to \infty} \frac{(h + 1)(\ln(l(h + 1)) + 1)}{l} = 0.
\]

That is

\[
\lim_{l \to \infty} \frac{G^A(l)}{l} = 0.
\]

So theorem 2 holds true.
Simulation Analysis

Calculating Skills

During the process of selecting support vectors, we decide that the sample whose distance to the subspace spanned by $x_h$ is bigger than $\epsilon$ should be added into $x_h$. We do this considering two reasons. The calculating precision of the computer is the first one. If $\epsilon = 0$, the matrix K in equation (8) may be close to singular or bad conditioned. The second reason is that a small $\epsilon$ has almost no effects on the squares error.

Simulation Result

We present an example to illustrate how effective the algorithm is. The training data are shown as Figure 1, where two classes are individually indicated by “•” and “□” (300 points for each class) in a two-dimension input space. The independent elements are denoted with “o” especially.

Figure 2 shows the simulating result. We could easily find that because many of the vectors are deleted from dataset, the number of the training data is much less than that of the whole dataset. Though the experiment is only in a two-dimension space, it could easily been generalized to high dimensional space.

![Figure 1. Scatter diagram of data set.](image1)

![Figure 2. Simulation result of classification based on VLS-SVM.](image2)
References