Steepest descent method based LSSVM model

Jinjin Liang
Department of Sciences, Xi’an Shiyou University, Xi’an 710071, China
myonlyonly@126.com

Abstract. A new least square support vector machine SPLSSVM is constructed in the primal space, and the steepest descent method is designed to figure out the optimal solution by defining the optimal condition as the energy of the system. We rewrite the objective function by replacing the two norms of the slack vector with the slack obtained from the equality constraints, and we derive an unconstrained optimization model. By setting gradient of the obtained objective function equal to zero, a linear system is derived. An energy function is defined and an interactive method is designed to figure out the optimal solution. The incomplete Cholesky factorization is used in the nonlinear space to approximate the kernel map before applying the steepest descent method. Numerical experiments demonstrate that the proposed SPLSSVM has higher precision and lower training time than SVM and LSSVM.

1. Introduction
Support Vector Machine (SVM) is proposed by Vapnik based on Structural Risk Minimization (SRM) principle [1], and it has been widely applied in pattern recognition, function estimation area. The training of SVM is characterized by convex constrained quadratic programming, and dual technique is used to figure out the optimal solution. Least Square Support Vector Machine has been proposed as a way to replace the quadratic programming problems by solving a linear system [2]. Although lacking theoretical analysis, LSSVM has been successfully applied to deal with many real problems due to its simplicity and faster training speed.

Existing researches focus on applying LSSVM to various areas [3,4,5], to solve approximate solutions for various differential equations. Reference [6] points out that the approximation solution in the primal space is superior to that in the dual space. Based on the research, some algorithms are designed in the primal space, such as the model selection method, or optimization algorithms for classification or regression. Inspired by this idea, we propose a LSSVM model in the primal space. Since LSSVM determines its solution by a linear system, we use the least square error to define a new objective function, and define it as the energy of the system [7, 8] at the current state. We then apply the steepest descent method to figure out the solution. In the linear space, we directly obtain the optimal solution. In the nonlinear space, we introduce the incomplete Cholesky factorization to obtain the explicit form of the nonlinear map.

2. Architecture of LSSVM
Let us consider a given binary training set \( \{(x_i, y_i)\}_{i=1}^{l} \) with input data \( x_i \in R^n \) and output data \( y_i \in \{-1,1\} \). The goal of classification is to construct a classifier of

\[
y(x) = \text{sign}(w^T x + b)
\]

in the linear space or
in the nonlinear space, where $w$ and $b$ are the weight and the bias of the separating hyperplane, and $\varphi: x \rightarrow \varphi(x)$ is the nonlinear map to transform data in the original space to a high feature space.

2.1. Linear LSSVM
LSSVM changes the inequality constraint of SVM into equality constraint, and is characterized by the following quadratic programming, where $\xi_i$ is the tolerance to the misclassified point $x_i$ or called the slack, $C > 0$ is the penalty proportional to the amount of violations $\sum_{i=1}^{l} \xi_i^2$.

$$\min \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{l} \xi_i^2$$

s.t. $y_i (w^T x_i + b) = 1 - \xi_i, i = 1, \cdots, l.$
Introducing a set of Lagrange multipliers $\alpha_i (i = 1, \ldots, l)$ for each constraint, we obtain

$$L(w, b, \xi, \alpha) = \frac{1}{2} ||w||^2 + \frac{C}{2} \sum_{i=1}^{l} \xi_i^2 - \sum_{i=1}^{l} \alpha_i [y_i (w \cdot \varphi(x_i) + b) - 1 + \xi_i]$$

By KKT condition, the partial derivative with respect to each variable equal to zero, and we obtain

$$A + \frac{1}{C} I \alpha = \begin{bmatrix} y & y \end{bmatrix}$$

$$y^T b = 0$$

where $A_{ij} = y_i y_j (x_i, x_j)$, $I$ is the identity matrix, $(x_i, x_j) = x_i^T x_j$ is the inner product of two vectors, $y = [y_1, \ldots, y_l]^T$ and $e = [1, \ldots, 1]^T$ are column vectors.

Denote the solution to (4) by $\alpha$ and $b$, and take the expression of $w = \sum_{i=1}^{l} \alpha_i y_i x_i$ in account, the classification function can be denoted by

$$y(x) = \text{sign} \left[ \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b \right]$$

2.2. Non Linear LSSVM
In the nonlinear space, use $\varphi: x \rightarrow \varphi(x)$ to map the training data into a high dimensional feature space $F$. Use the same denotations as before, LSSVM is formulated as follows.

$$\min \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{l} \xi_i$$

s.t. $y_i (w^T \varphi(x_i) + b) = 1 - \xi_i, i = 1, \cdots, l.$

Similarly, we derive the above optimality condition in the following linear systems

$$\Omega + \frac{1}{C} I \alpha = \begin{bmatrix} y & y \end{bmatrix}$$

$$y^T b = 0$$

where $\Omega_{ij} = y_i y_j K(x_i, x_j)$ with $\Omega_{ij} = y_i y_j k(x_i, x_j)$ and $k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ is the kernel function; $I$ is the identity matrix, $y = [y_1, \ldots, y_l]^T$ and $e = [1, \cdots, 1]^T$ are column vectors.

Figuring out the solution $\alpha$ and $b$ to the above linear systems, LS-SVM constructs a classifier of

$$y(x) = \text{sign} \left[ \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b \right]$$
3. SPLSSVM in Primal Space

3.1. Model derivation

Based on the LSSVM in the linear space in the above section, we rewrite the slack in the equality constraint as

\[ \xi_i = 1 - y_i(w^T x_i + b) \]  

(10)

Multiply \( y_i \) to both sides of the above equation, we have

\[ y_i \xi_i = y_i - (w^T x_i + b) \]  

(11)

Apply square operation to both sides of the above equation, we have

\[ (\xi_i)^2 = (y_i \xi_i)^2 = [(w^T x_i + b) - y_i]^2 \]  

(12)

and we substitute the equality constraint into program (1), the linear primal LSSVM is obtained

\[ \min F(w, b) = \frac{1}{2} w^T w + \frac{C}{2} [Aw + be - y]^T [Aw + be - y] \]  

(13)

where \( A \) is the input data matrix.

For the nonlinear LSSVM model, we deduce the square of the slack vector in the same way.

\[ (\xi_i^2) = (y_i \xi_i)^2 = [(\varphi(x_i) + b) - y_i]^2 \]  

(14)

Substitute the least square error into the objective function, the program (7) can be formulated as

\[ \min F(w, b) = \frac{1}{2} w^T w + \frac{C}{2} [\varphi(A)w + be - y]^T [\varphi(A)w + be - y] \]  

(15)

In the above section, \( \varphi(A) \) is the transformed matrix of input data \( A \). Making use of the kernel trick, we don’t have to know the explicit form of the nonlinear map. Since the kernel matrix usually is symmetric positive definite or symmetric semi-positive definite, we apply the incomplete Cholesky factorization to \( K = \varphi(A)^T \varphi(A) \), and obtain the approximate expression of \( \varphi(A) \).

Denote \( \alpha = [w, b]^T \), then the unconstrained programming in the two spaces can be formulated as the standard quadratic programming form.

\[ \min F(\alpha) = \frac{1}{2} \alpha^T Q \alpha - d^T \alpha \]  

(16)

where \( Q = \begin{bmatrix} A^T A + \frac{I}{C} & A^T e \\ e^T A & l \end{bmatrix} \) and \( d = \begin{bmatrix} -A^T y \\ -e^T y \end{bmatrix} \) corresponds to the model in the linear space; and

\[ Q = \begin{bmatrix} \varphi(A)^T \varphi(A) + \frac{I}{C} & \varphi(A)^T e \\ e^T \varphi(A) & l \end{bmatrix} \]  

\( d = \begin{bmatrix} -\varphi(A)^T y \\ -e^T y \end{bmatrix} \) corresponds to the model in the nonlinear space.

According to the optimization condition, we set the partial derivative of the objective function to zero, and obtain the following linear system

\[ \nabla F(\alpha) = Q \alpha - d = 0 \Rightarrow Q \alpha = d \]  

(17)

3.2. Steepest descent method

Denote by \( \alpha \) the solution to (19), we define the following function with \((\cdot, \cdot)\) as the inner product

\[ F(y) = \frac{1}{2} (y - \alpha, Q(y - \alpha)) \]  

(18)

It is obvious that the function \( F(y) \) has a unique minimum at \( y = \alpha \).

Define the energy function as
\[ E(y) = F(y) - F(0) = \frac{1}{2}[(y, Qy) - (y, d)] \]  

(19)

If the value of \( E(y) \) can be decreased at each step until \( \alpha \) is obtained for which \( E(y) \) is minimal or nearly minimal. In many applications, the function \( E(y) \) represents a quantity of significance, such as the energy of the system. In these cases the solution is the state of minimum energy.

The gradient of the function \( E(y) \) is

\[ G(y) = Qy - d = -r \]  

(20)

We use the steepest descent method, starting form an initial vector \( \alpha^0 \), and update as

\[ \alpha^{k+1} = \alpha^k + \beta_k r^k \]  

(21)

where \( r^k = d - Qx^k \) and \( \beta_k \) is the step parameter.

To choose the step parameter, we use the following rule so that \( E(\alpha^{k+1}) \) is minimal.

\[
E(\alpha^{k+1}) = E(\alpha^k + \beta_k r^k) = \frac{1}{2} (\alpha^k, Q\alpha^k) + \beta_k (r^k, Q\alpha^k) + \frac{1}{2} \beta_k^2 (r^k, Qr^k) - (\alpha^k, d) - \beta_k (\alpha^k, d) \\
= E(\alpha^k) - \beta_k (r^k, r^k) + \frac{1}{2} \beta_k^2 (r^k, Q\alpha^k)
\]  

(22)

This expression is a quadratic function in \( \alpha^k \) and has a minimus for some value of \( \alpha^k \). By setting the partial derivative to zero, we can find that

\[
\alpha^k = \frac{(r^k, r^k)}{(r^k, Qr^k)} = |r^k|^2 / (r^k, Qr^k)
\]  

(23)

\[
(r^{k+1}, r^k) = (r^k, r^k) - \beta_k (r^k, Qr^k) = 0
\]  

(24)

Showing that consecutive residuals are orthogonal, for this optimal choice of \( \beta_k \), we have

\[
E(\alpha^{k+1}) = E(\alpha^k) - \frac{1}{2} |r^k|^2 / (r^k, Qr^k)
\]  

(25)

The energy function will decrease as \( k \) increases until the residual is zero. Notice from the definitions of \( E(\alpha^k) \) and \( r^k \) that

\[
E(\alpha^k) = \frac{1}{2} (Q^{-1}r^k, r^k) - F(0)
\]  

(26)

Hence the formula is equivalent to

\[
(Q^{-1}r^{k+1}, r^{k+1}) = (Q^{-1}r^k, r^k) - |r^k|^2 / (r^k, Qr^k)
\]  

(27)

The formulas for this steepest gradient method is

\[
x^{k+1} = x^k + \beta_k r^k \\
r^{k+1} = r^k - \beta_k Qr^k
\]  

(28)

\[
\beta_k = |r^k|^2 / (r^k, Qr^k)
\]

The following theorem proves that the above method converges for any initial iterate \( \alpha^0 \).

**Theorem 1.** The steepest descent method converges to the unique solution for any initial iterate \( \alpha^0 \).

**Proof.** First note that \( Q \) is symmetric and positive definite, as can be easily seen from
\[ Q = \begin{bmatrix} \varphi(A)^T \varphi(A) + \frac{I}{C} \varphi(A)^T e \\ e^T \varphi(A) \\
\end{bmatrix} \]

(29)

The inverse matrix \( Q^{-1} \) is positive definite, and so is \( Q^T Q^{-1} \), we have there are constants \( c_0, c_1 \) such that for all vectors \( \alpha \)

\[ c_0 (\alpha, Q^{-1} \alpha) \leq (\alpha, Q^T Q^{-1} \alpha) \]

\[ c_1 (\alpha, Q \alpha) \leq (\alpha, \alpha) \]  

(30)

Note

\[ (r^{k+1}, Q^{-1} r^{k+1}) = (r^k, Q^{-1} r^k) - \beta_k (r^k, r^k) - \beta_k (Q r^k, Q^{-1} r^k) + \beta_k^2 (r^k, Q r^k) \]

\[ = (r^k, Q^{-1} r^k) - \beta_k (r^k, Q^T Q^{-1} r^k) \]  

(31)

Now using \( c_1 (\alpha, Q \alpha) \leq (\alpha, \alpha) \), we have

\[ \beta_k \leq \frac{(r^k, r^k)}{(r^k, Q r^k)} \geq c_1 \]  

(32)

And thus we have for \( k \geq 0 \)

\[ (r^{k+1}, Q^{-1} r^{k+1}) \leq (r^k, Q^{-1} r^k)(1 - c_0 c_1) \]  

(33)

Notice that \((1 - c_0 c_1)\) is nonnegative, since \( Q^{-1} \) is positive definite. Therefore,

\[ (r^k, Q^{-1} r^k) \leq (r^0, Q^{-1} r^0)(1 - c_0 c_1)^k \]  

(34)

And thus \((r^k, Q^{-1} r^k)\) tends to zero.

Since \( r^k = d - Q x^k \) and \( Q^{-1} \) is positive definite, the vectors \( r^k \) converge to zero and

\[ x^k = Q^{-1} (d - r^k) \]  

(35)

It follows that the vectors \( x^k \) converge to \( Q^{-1} d \), which is the unique solution to program (16).

4. Numerical Experiments

We demonstrate now the effectiveness and speed of LSSVM in the primal space on several datasets. All the experiments are carried out on a PC with P4 CPU, 3.06 GHz, 1GB Memory. The programs are written in pure MATLAB 7.01 Language.

Example 1 Generate 500 normal distribution data of two classes, where the positive and negative samples are of the same size. We randomly select 250 as the training set, and treat the others as the testing set. Using linear kernel function \( K(x, y) = x^T y \), the performances of the proposed SPLSSVM and LSSVM are as follows.

| Penalty | Positive Precision | Negative Precision | Train Time | Test Time |
|---------|--------------------|--------------------|------------|-----------|
| 0.1     | 99.38              | 98.74              | 1.35       | 1.23      |
| 0.3     | 95.68              | 100                | 1.37       | 1.92      |
| 0.5     | 100                | 99.65              | 1.43       | 1.61      |
| 1       | 100                | 98.54              | 1.15       | 1.78      |

The following solution is obtained from table 1.

(1) SPLSSVM has high training and testing accuracies, whose training accuracies can achieve 100% and the testing accuracies can achieve 99.65%.
(2) SPLSSVM has low training and testing time, the training time is approximately 1.35 second, and the testing time is approximately 1.35 second.

**Example 2** Pima Indians

Pima Indians is a UCI dataset composed of 500 plus examples and 268 minus examples, eight attributes for each data. Ten-fold cross validation is used to choose the optimal parameters. The Radial Basis Kernel function is used, where the penalty and the kernel width are respectively selected from $C \in \{0.01, 0.1, 0.5, 0.8, 1, 5, 10, 15\}$ and $\sigma \in \{0.01, 0.02, 0.05, 0.1, 0.5, 0.8, 1, 1.5\}$.

|          | SPLSSVM | SVM  | LSSVM |
|----------|---------|------|-------|
| TRP.     | **79.18** | 78.82 | 74.33 |
| TEP.     | **80.27** | 78.36 | 78.27 |
| TRT.     | **6.61**  | 21.45 | 7.35  |
| TET.     | **5.31**  | 17.55 | 6.13  |

In the above table, ‘TRP’ stands for training precision and ‘TEP’ stands for testing precision; ‘TRT’ stands for training time and ‘TET’ stands for testing time. The highest accuracies and the lowest training time are illustrated in bold type numbers. It is obvious to see that, SPLSSVM has a precision comparable with that of SVM, and higher than that of LSSVM; SPLSSVM has a training time much lower than that of LSSVM and SVM.

5. **Conclusions**

A least square model LSSVM in the primal space is derived, by directly substituting the equality constraint into the objective function. We see the optimization condition as the energy at different state, and design the steepest descent method to figure out the solution. Various experiments on artificial data and UCI data demonstrate that the proposed SPLSSVM has higher accuracies than traditional LSSVM, and has a much lower training time than that of SVM. Further research includes designing efficient energy schemes to reduce the current energy state.

**Acknowledgments**

This work was financially supported by the Natural Science Foundation of Shaanxi Educational Commission (2016JK1596).

**References**

[1] Zou Bin, Peng Zhiming, Xu Zongben. The learning performance of support vector machine classification based on Markov sampling[J]. Science china information sciences, 2013, 56(3): 1-16.

[2] Pijush1 Samui, Tim Lansivaara, Madhav R. Bhatt. Least square support vector machine applied to slope reliability analysis [J]. Geotechnical and Geological Engineering, August 2013, 31(4): 1329-1334

[3] Mehrkanoon S , Suykens J A K . Learning solutions to partial differential equations using LS-SVM[J]. Neurocomputing, 2015, 159:105-116.

[4] Si Gangquan, Lou Yong, Zhang Yinsong. Robust Least Squares Support Vector Machines with Applications to Soft-Sensing. Journal of Xi’an Jiaotong University, 2012, 46(8): 15-21.

[5] Zhang Guoshan, Wang Yiming, Wang Shiwei, Liu Wanquan. Improved method to solve ordinary differential equations approximate solutions based on LS-SVM. Journal of Systems Science and Mathematical Sciences, 2013, 33(6): 695-707.

[6] Moore, G. Model selection for primal SVM [J]. Machine Learning, 2011, 85(1-2): 175-208

[7] Xiaofeng Yang. Linear, first and second-order, unconditionally energy stable numerical schemes for the phase filed model of homopolymer blends. Journal of Computational Physics, 2016,
[8] Hongwei Li, Lili Ju, Chenfei Zhang. Unconditionally Energy Stable Linear Schemes for the Diffuse Interface Model with Peng-Robinson Equation of State. Journal of Science Computing, 2018, 38(6): 993-1015.