Application of a Sparse Least Squares Support Vector Machine Algorithm in Radar Target Recognition

Zhao Dongbo*[1], Li Hui[2]

[1]School of Electronic Engineering, Xi'an Aeronautical University, Xi’an,710077,China
[2]School of Electronic Information, Northwestern Polytechnical University, Xi’an,710129,China
Email: alien_dffy@163.com

Abstract. Least squares support vector machine (LS-SVM) has a large amount of computation and sparsity. Aiming at this problem, a fast sparse approximation least squares support vector machine (FSALS-SVM) algorithm is proposed. The algorithm uses an iterative algorithm of complexity to accelerate the calculation of the inverse of the kernel matrix and the sparse processing of the support vector machine through the pruning algorithm, thus reducing the computational complexity. The classification and recognition experiments of one dimension range profile of one dimensional radar target show that the FSALS-SVM is more sparsely under the same generalization performance.

Keywords: Least Squares Support Vector Machine (LS-SVM); Fast Sparse Approximation Least Squares Support Vector Machine (FSALS-SVM); pruning algorithm; Kernel Principal Component Analysis (KPCA); Central moment; Radar Target Recognition

1 Introduction

The kernel machine learning method is based on statistical learning theory and kernel method. It is a statistical machine learning method based on support vector machine. Because the structural risk criteria in the statistical learning theory are adopted in the nuclear machine, the nuclear machine will not fall into the over fitting, so it has good generalization ability. It does not require a lot of training samples. To a certain extent, it solves the problem of over-fitting and the extension ability of the neural network. At the same time, because of the adoption of the nuclear method, the nonlinear nuclear machine is realized, which makes the nuclear machine have a larger hypothesis space, can solve the more complex problems, and does not need too large calculation. This also avoids the dimension disaster problem in the neural network to some extent.

Support Vector Machine (SVM) is a new pattern recognition method developed on the basis of statistical learning theory. It is based on the VC dimension theory of statistical learning theory and the principle of structural risk minimization. It can avoid the local optimal solution, overcome the "dimension disaster", and show many unique advantages in solving the problem of small sample, nonlinear and high dimensional pattern recognition.

The SVM algorithm transforms the actual problem into the high dimensional feature space through nonlinear transformation, constructs a linear discriminant function in the high dimensional space to realize the nonlinear discriminant function in the original space, and solves the dimension problem skillfully. The complexity of the algorithm is independent of the dimension of the sample. SVM has the characteristics of simple structure, good generalization ability, unique solution, suitable for processing high dimensional data, computational complexity and dimensionality of training samples. Therefore, it is an effective method for the model classification problem of high dimension and small sample. For the recognition of radar target high resolution range image (HRRP), SVM is undoubtedly an effective means. SVM can obtain higher correct recognition rate and better generalization performance.

2 Support Vector Machine (SVM)

2.1 SVM Principle

The support vector machine implements the following idea: the input vector x is mapped to a high dimensional feature space Z by selecting a good nonlinear mapping in advance, and the optimal hyperplane is constructed in this space. The so-called optimal classification surface is that the classification surface can not only make two kinds of sample points without fault, but also make the two categories of the largest gap. As shown in the optimal hyperplane. (As shown by the formula 1), the hollow circle and the solid circle in the graph represent two classes of training samples respectively. H is the classification line that separates the two classes correctly. H1 and H2 are the straight lines passing through the nearest samples from the classification lines and parallel to the classification lines, and the distance between them is called the classification interval (margin). The former is to ensure the minimum empirical risk, and the latter is actually to minimize the confidence range and minimize the actual risk, which is a concrete embodiment of the principle of structural risk minimization. Generalized to the higher dimension, the optimal classification line becomes the optimal hyperplane.

The general form of the linear discriminant function in the D dimensional space is that the classification surface equation is the normalization of the discriminant function, which makes all the samples of the two classes satisfied, at this time, from the nearest sample of the classification surface, and that the classification is correctly classified in the face of all the samples, which is required to satisfy it.

$$y_i(w^T x_i + b) - 1 \geq 0, i = 1, 2, \cdots, n \tag{1}$$

In the formula (1), the samples that equate the numbers are called Support Vectors.

The classification gap (Margin) interval between two types of samples is:

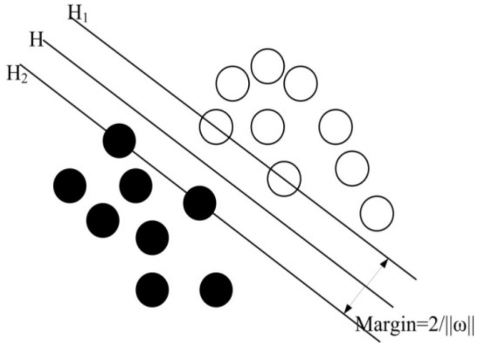

Figure 1. Optimal Hyperplane

$$M\arg in = 2 / \left\| w \right\| \tag{2}$$

Therefore, the optimal classification surface problem can be expressed as the following constraint optimization problem, that is, under the constraint of condition formula (1), obtain the minimum value for the function

$$\varphi\left(w\right) = \frac{1}{2}\left\| w \right\|^2 = \frac{1}{2}(w^T w) \tag{3}$$

To this end, you can define the following Lagrange functions:

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i [y_i (w^T x_i + b) - 1] \tag{4}$$

Among them, for the Lagrange coefficient, our problem is to find the minimum value of Lagrange function for W and B. The formula (4) pairs W, B, and partial differential respectively, and makes them equal to 0.

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{5}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{6}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow \alpha_i [y_i (w^T x_i + b) - 1] = 0 \tag{7}$$

The above three expressions plus the original constraint condition can transform the original problem into the following two problems:

$$\begin{cases} \max \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(x_i^T x_j \right) \\ s.t \qquad a_i \geq 0, i = 1, \cdots, n \\ \qquad \sum_{i=1}^{n} a_i y_i = 0 \end{cases} \tag{8}$$

This is a quadratic function problem with inequality constraints, and there exists a unique optimal solution. If it is the optimal solution, then

$$w^* = \sum_{i=1}^{n} a_i^* y_i x_i \tag{9}$$

The sample which is not zero is called the support vector. Therefore, the weight vector of the optimal classification surface is a linear combination of support vectors.

b* can be solved by constraint conditions, and the optimal classification function obtained is:

$$f\left(x\right) = \mathrm{sgn}((w^*)^T x_i + b^*) = \mathrm{sgn}(\sum_{i=1}^{n} a_i^* y_i x_i^* x + b^*) \tag{10}$$

SGN () is a symbolic function.

When a hyperplane cannot be completely separated from the two points, the relaxation variable can be introduced to satisfy the hyperplane.

$$y_i (w^T x_i + b) \geq 1 - \xi_i \tag{11}$$

When the value is between 0 and 1, the sample point Xi can still be correctly classified, and when the value is greater than 1, the sample point Xi is misclassified. To this end, the following objective functions are introduced:

$$\psi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{12}$$

C is a normal number, which is called a penalty factor. At this point, SVM can be realized by two times programming (dual programming).

$$\begin{cases} \max \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(x_i^T x_j \right) \\ s.t \qquad 0 \leq a_i \leq C, i = 1, \cdots, n \\ \qquad \sum_{i=1}^{n} a_i y_i = 0 \end{cases} \tag{13}$$

2.2 Kernel Function of Support Vector Machine (SVM)

If the simple hyperplane in the original space cannot be satisfactorily classified, the complex hypersurface must be used as the interface.

First, the input space is transformed into a high dimensional space by nonlinear transformation, and then the optimal linear classification surface is obtained in the new space, which is realized by defining an appropriate kernel function (inner product function).

$$K(x_i, x_j) = \left\langle \Phi(x_i) \cdot \Phi(x_j) \right\rangle \tag{14}$$

Instead of the dot product in the optimal classification plane, the kernel function is equivalent to transforming the original feature space into a new feature space.

$$Q\left(a\right)=\sum_{i=1}^{n}a_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K\left(x_i,x_j\right) \tag{15}$$

The corresponding discriminant function is as follows:

$$f\left(x\right)=\operatorname{sgn}[(w^*)^T\varphi(x)+b^*]=\operatorname{sgn}(\sum_{i=1}^{n}a_i^* y_i K(x_i,x)+b^*) \tag{16}$$

It is a support vector, as an unknown vector, formula (16) is SVM, which is similar to a neural network in the form of a classification function, and its output is a linear combination of a number of middle layer nodes, and each middle layer node corresponds to the inner product of the input sample and a support vector, and therefore is also called the support vector network.

Since the final discriminant function actually contains only the linear combination of the inner product of the unknown vector and the support vector, the computational complexity of the recognition depends on the number of the support vectors.

At present, there are mainly three types of kernel functions, which are related to the existing algorithms. (1) polynomial form of kernel function, that is,

$$K\left(x,x_i\right)=\left[\left(x^Tx_i\right)+1\right]^q \tag{17}$$

corresponding to SVM is a Q order polynomial classifier.
(2) radial basis function kernel function

$$K\left(x,x_i\right)=\exp\{-\frac{\left\|x-x_i\right\|^2}{\sigma^2}\} \tag{18}$$

corresponding to SVM is a radial basis function classifier.
(3) S - shaped kernel function, such as

$$K\left(x,x_i\right)=\tanh(v(x^Tx_i)+c) \tag{19}$$

then SVM is a two layer perceptron neural network, but not only the weight value of the network, but also the number of hidden layer nodes of the network is also automatically determined by the algorithm.

3 Least Squares Support Vector Machine (LS_SVM)

3.1 LS_SVM Algorithm Derivation

For a given set of training samples and corresponding class tag sets, the training sample X is mapped from the original input space to a high dimensional feature space F, that is, through a nonlinear mapping. At this time, the input training sample is transformed from the original X to the sample set in the D dimension space, and the high dimensional sample set is obtained after the nonlinear mapping to the high dimensional feature space F. The goal of the least squares support vector machine model is to construct a classifier in the following form:

$$y=\operatorname{sgn}(w^T\varphi(x)+b) \tag{20}$$

So that the sample x can be correctly classified by function g (x), where SGN () is a sign function. Using LS-SVM algorithm to find the optimal classification hyperplane in F of high-dimensional feature space is to solve the following optimization problems:

$$\begin{aligned} &y=\operatorname{sgn}(w^Tw+\frac{\gamma}{2}\sum_{i=1}^{l}e_i^2)\\ &s.t\ \ y_i=w^T\varphi(x_i)+b+e_i\ \ (i=1,\ldots,l)\end{aligned} \tag{21}$$

It actually plays the role of controlling the degree of punishment for the wrong sample. If the training data contains large noise, it should be smaller. From the point of view of statistical learning theory, the first item of formula (21) can be regarded as the confidence range, which affects the complexity of the learning machine; the second item can be considered as an empirical risk.

In order to obtain the minimum value of the objective function (formula 21), we now construct the Lagrange function L as follows:

$$L(w,b,\xi,\alpha) = J(w,\xi) - \sum_{i=1}^{l} \alpha_i \left\{ y_i \left[w^T \phi(x_i) + b \right] - 1 + \xi_i \right\} \tag{22}$$

This is the Lagrange multiplier coefficient, also known as the support value, because it is equality constraint, so its value can be positive and negative. In order to find the minimum value of the Lagrange function L, we present the following optimization conditions for the function L for the variable, the sum, and the partial derivative, and the partial derivative equal to 0.

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \tag{23}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l} y_i \alpha_i = 0 \tag{24}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow r\xi_i = \alpha_i \Rightarrow \xi_i = \frac{\alpha_i}{r} \tag{25}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow y_i \left[w^T \phi(x_i) + b \right] - 1 + \xi_i = 0 \tag{26}$$

The formula (23) and (25) substitution (26) eliminate and obtain:

$$\sum_{j=1}^{l} \alpha_j y_i y_j k(x_i, x_j) + \frac{1}{r}\alpha_i + b y_i = 1 \tag{27}$$

As a result, the type (27) can be extended into a set of linear equations composed of L equations, and formula (27) is only the I equation in the linear equation group. Taking the Lagrange multiplier vector and parameters as variables, we can write the matrix (24) and (27) into the following matrix form:

$$\begin{bmatrix} 0 & Y^T \\ Y & A \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \overline{1} \end{bmatrix} \tag{28}$$

Here, I will focus on the composition of matrix A:

$$A = \Omega + \frac{1}{r} I \tag{29}$$

I is the unit matrix of order L. In order to describe it conveniently, we can call matrix A as kernel correlation matrix. The solution of the equation group (28) can be obtained:

$$b = \frac{Y^T A^{-1} \overline{1}}{Y^T A^{-1} Y} \tag{30}$$

$$\alpha = A^{-1}(\overline{1} - Yb) \tag{31}$$

Generally speaking, it is a 1.5 positive definite symmetric matrix, and its inverse is very likely to not exist, but if each element is added to the diagonal, the inverse of the kernel correlation matrix A is in the general case, which is also the bright point of the least squares support vector machine.

Finally, the classification function of least squares support vector machines can be obtained:

$$f(x) = \operatorname{sgn}\left(w^T \varphi(x_i) + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i k(x, x_i) + b\right) \tag{32}$$

3.2 The Selection Method of Parameters

One of the advantages of SVM is that its parameters are very few, including only the selection of kernel function parameters and penalty factor gamma. The difference of kernel function mainly affects the complexity of the distribution of sample data in high dimensional feature space, and the role of penalty factor gamma is to adjust the confidence range of the learning machine and the proportion of empirical risk in the definite feature space. But how to take the appropriate parameters of the kernel function and the penalty factor - gamma, there is no definite guidance on the theory at present, which is usually determined by the trial and error method. This is a problem that needs further study.

At present, the commonly used parameter selection method is mainly the cross validation method (cross validation) in data mining theory. Cross validation is a model evaluation method, which can predict unknown data through learning. The fitting accuracy reflects the accuracy of the model to the historical statistical data fitting, reflecting the suitability of the past change rules, and the prediction accuracy reflects the prediction ability of the model to the future. A model has high fitting accuracy, and it has high prediction accuracy. In order to solve this problem, cross examination method is proposed. Cross examination is to divide training samples into M parts, one of which is retained as part of the training, and the remaining M-1 as the training part. The training part is trained to estimate the probability PR, and the remaining part is used as a test set to test the accuracy of the model.

3.2.1 Common Cross Test Method

The common cross test methods include:

(1) K-fold cross validation. The training samples are randomly divided into K disjoint subsets, and the sizes of each fold are approximately equal. K-1 training subsets are used to establish a set of parameters for a given set of parameters, and the performance of the parameters is evaluated using the remaining MSE of the last subset. According to the above process, repeat K times, and estimate the expected generalization error according to the average MSE value obtained after K iterations, and finally select a set of optimal parameters. The advantages of this approach are concerned with how data are divided, and each data point can only participate in one test, and the resulting prediction variance increases with the decrease of K. The disadvantage of this method is the large amount of computation and training.

(2) leave-one-out cross validation, which is actually a cross test, that is, a set of data with a K sample, each taking K-1 for training to predict the remaining samples. This method is deterministic and does not need to use random functions as a cross test to select the division of a training set. However, this method is too costly to calculate, and it is not stratified, that is, the ratio of the training set to the test set is different, and the result is sometimes very high.

(3) the holdout cross validation dichotomy cross test. It is the simplest type of cross validation, which is divided into two independent subsets, one part as a training set, and the other as a test set to estimate the prediction accuracy of the model. The cross test method requires more samples, and the estimation accuracy is often related to the data grouping, and the cross test method is not completely feasible in practical application.

3.2.2 Parameter Determination Method Based on Grid Search

The advantage of grid search method is that it can simultaneously search kernel parameters and penalty factor gamma two parameters. The specific steps are as follows: first, the parameters of the kernel function and the penalty factor gamma are selected, and then the search step is set. In this way, a two-dimensional network is formed on the kernel function parameters and the penalty factor gamma coordinate system, which corresponds to each set of values on the grid and calculates the prediction accuracy by the contour line, and the contour map is obtained. Accordingly, the contour map is obtained, and the best one is determined. Parameters. If the accuracy rate can not meet the requirement, a search area is selected based on the existing contour map, and the search step is reduced to fine search.

3.2.3 Parameter Determination Based on Evolutionary Algorithm

Evolutionary algorithm is a series of search techniques based on the basic computing model of natural evolution process. It is widely used in many fields, such as function optimization recognition, machine learning, neural network training, intelligent control and so on. Evolutionary algorithm (evolutionary algorithm) is a mature global optimization method with high robustness and wide applicability. It has the characteristics of self-organizing and self-adaptive, which can not be restricted by the nature of the problem. Commonly used methods are genetic algorithm (GA) or particle swarm optimization (PSO).

4 Improved Fast Sparse Least Squares Support Vector Machine

4.1 The Sparsity of Least Squares Support Vector Machines

The traditional SVM classifier has sparsity, that is, the classification hyperplane is only related to the less support vectors in the sample point. Even if all the non support vectors are removed, it does not affect the solution. This sparse feature can reduce computation greatly in the two learning process, and has important algorithm advantages.

The difference between the LS_SVM and the standard SVM method is that the least squares support vector machine modifies the empirical risk in the objective function into the sum of the squares of the classification error. Thus, the two planning problems in the standard support vector machine are skillfully converted to a linear formula, which makes it difficult to solve the optimal classification plane. The degree is greatly reduced. Another difference is that LS_SVM uses equality constraints to replace inequality constraints in standard support vector machines. However, the Lagrange multiplier for each sample of least squares support vector machine is rarely 0. In this way, all the sample points should be recalculated for the new sample points, so all training samples will always become support vectors. Therefore, compared with the general SVM method, the solution of LS_SVM does not have sparsity [1].

At present, sparse research of training samples is a hot and difficult point in the research field of least squares support vector machines. In solving the sparsity of least squares support vector machines, it is proposed to get sparsity by cutting a small number of smaller samples each time; Kruif and Vries are pruning methods that are removed by the samples with smaller approximation errors introduced after the deletion; and a pruning party based on SMO is proposed. Jiao and Bo et al. put forward a fast thinning method, which iteratively constructs a decision function by adding a centralization to the base function of the training sample, until the epsilon insensitive criterion [1] is satisfied.

In order to overcome the two shortcomings of least squares support vector machine, a fast sparse least squares support vector machine is presented.

4.2 Iterative Inversion of Kernel Matrix

It is the most time-consuming to calculate the inverse of the kernel matrix in the LS_SVM algorithm. If the inverse of the kernel matrix is calculated from zero at every iteration step, the computational complexity is $O(n^3)$, which will result in a very high computing complexity. Therefore, an iterative algorithm with complexity $O(n^2)$ is introduced to speed up the computation of the inverse of the kernel matrix. To facilitate the description, we will rewrite the formula (22) as follows:

$$\min\left(f=\frac{1}{2}\begin{bmatrix} b & \alpha^T \end{bmatrix}\begin{bmatrix} 0 & 1^T \\ 1 & K^T \end{bmatrix}\begin{bmatrix} b \\ \alpha \end{bmatrix}-\begin{bmatrix} b & \alpha^T \end{bmatrix}\begin{bmatrix} 0 \\ y \end{bmatrix}\right) \tag{33}$$

And, 1 is a unit matrix.

If we choose the first S basis function in the (n+1) generation, we have:

$$R^{n+1}=\begin{pmatrix} 0 & 1^T & 1 \\ 1 & \overline{K}_{PP} & \overline{K}_S \\ 1 & \overline{K}_S^T & \overline{K}_{SS} \end{pmatrix}^{-1} \tag{34}$$

Among them, $\overline{K}_S=\left[\overline{K}_{P_1S},\overline{K}_{P_2S},\ldots,\overline{K}_{P_nS}\right]^n$. Suppose that in the N generation they have been calculated

$$R^n=\begin{pmatrix} 0 & 1^T \\ 1 & \overline{K}_{PP} \end{pmatrix}^{-1} \tag{35}$$

According to the inverse matrix lemma below, the matrix R^{n+1} can be obtained at the cost of computation $O(n^2)$.

Lemma 1: gives a reversible matrix A and matrix U, V, D, equality is established.

$$\begin{pmatrix} A & U \\ V & D \end{pmatrix}^{-1}=\begin{pmatrix} A^{-1}+A^{-1}U(D-VA^{-1}U)^{-1}VA^{-1} & -A^{-1}U(D-VA^{-1}U)^{-1} \\ -(D-VA^{-1}U)^{-1}VA^{-1} & (D-VA^{-1}U)^{-1} \end{pmatrix} \tag{36}$$

By substituting formula (36) in formula (35) to solve the inverse matrix, we can get the following update formula:

$$R^{n+1}=\begin{pmatrix} R^n & 0 \\ 0^T & 0 \end{pmatrix}+\lambda\begin{pmatrix} \beta \\ -1 \end{pmatrix}\begin{pmatrix} \beta^T & -1 \end{pmatrix} \tag{37}$$

Among them, $\lambda = \left(\overline{K}_{SS} - \begin{bmatrix} 1 & \overline{K}_S^T \end{bmatrix} \beta \right)^{-1}$. According to the formula $\beta = R^n \begin{pmatrix} 1 \\ \overline{K}_S \end{pmatrix}$, the iterative calculation formulas for a and B in the N generation are as follows:

$$\begin{bmatrix} b^{n+1} \\ \alpha_p^{n+1} \\ \alpha_s^{n+1} \end{bmatrix} = R^{n+1} \begin{bmatrix} 0 \\ y_p \\ y_s \end{bmatrix} = \begin{bmatrix} R^n \begin{bmatrix} 0 \\ y_p \end{bmatrix} \\ 0 \end{bmatrix} + \lambda \left(\beta^T \begin{bmatrix} 0 \\ y_p \end{bmatrix} \quad -y_s \right) \begin{bmatrix} \beta \\ -1 \end{bmatrix} \tag{38}$$

Suppose that the A and B have been calculated using the equation $\begin{bmatrix} b^{n+1} \\ \alpha_p^n \end{bmatrix} = R^n \begin{bmatrix} 0 \\ y_p \end{bmatrix}$ in the N generation. there are

$$\begin{bmatrix} b^{n+1} \\ \alpha_p^{n+1} \\ \alpha_s^{n+1} \end{bmatrix} = \begin{bmatrix} b^n \\ \alpha_p^n \\ 0 \end{bmatrix} + \lambda \left(\beta^T \begin{bmatrix} 0 \\ y_p \end{bmatrix} \quad -y_S \right) \begin{bmatrix} \beta \\ -1 \end{bmatrix} \tag{39}$$

Formulas (37) and (38) show that we can avoid direct computation of the inverse matrix, R, α and b are updated effectively, and the computational complexity is only $O(n^2)$.

4.3 Pruning Algorithm Sparsity

The classification function of standard least squares support vector machine (the optimal classification surface function) is:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i k(x, x_i) + b\right) \tag{40}$$

By observing the optimal classification surface function above, we can find that the support value α_i has the greatest influence on the classification plane. The smaller the absolute value of the support value α_i, the smaller the contribution to the taxonomy, that is, the support value of the smaller absolute value has little effect on the classification. In this case, then the training samples of the support values of the smaller absolute values can be deleted. At present, the common least squares support vector machine sparsity algorithm is generally based on the above thought. First, the training sample of a certain scale is selected, and then a certain method is adopted to delete the training samples with relatively small absolute value of the support value in the learning process, so as to achieve the purpose of sparsity.

In this paper, the pruning algorithm is used to sparse support vector machines. The basic idea of the pruning algorithm is: After the $|\alpha_i|$ descending order is arranged, the map of support vector value is obtained. As the size of $|\alpha_i|$ reflects the relative importance of each training sample in the composition of the solution vector, the training samples with smaller $|\alpha_i|$ can be deleted. Deleting a training sample is equivalent to zeroing its corresponding support value, that is $\alpha_i = 0$. In this way, a better sparsity can be obtained by continuously removing training samples from training samples which have less influence on decision or regression function and training the newly acquired training set.

The pruning algorithm is similar to the algorithm used in neural network to delete hidden nodes, but there is no need to solve the inverse matrix of the Hessian matrix. It only needs to zero the support value of the smaller absolute value in the resolution vector, so the algorithm is simple to execute. Combining the iterative inversion of kernel functions, the fast sparse LS-SVM algorithm proposed in this chapter can be summarized as follows:

Step 1. Let $n = 0, P = \varnothing$;

Step 2. select the support vector: $R^{n+1} = \begin{bmatrix} 0 & 1 \\ 1 & k(x_s, x_s) + \dfrac{1}{2\gamma} \end{bmatrix}^{-1}, \begin{bmatrix} b^{n+1} \\ \alpha_s^{n+1} \end{bmatrix} = R^{n+1} \begin{bmatrix} 0 \\ y_S \end{bmatrix}, n = n + 1$

If n=0, using the classifier obtained in the previous step to classify the samples which are not selected to get f (x), then calculate the value of the function $g(x) = yf(x)$, get the label V of the minimum value, update the set $p_{n+1} = p_n + \{v\}$, and then calculate $R^{n+1}, \alpha_p^{n+1}, \alpha_s^{n+1}, b^{n+!}$ according to the formula (35) and (39), and finally update the n=n+1;
Step 3: In order to better control the scale and improve the performance, performs the pruning operation when the two conditions of the incremental process iterates the number of L times and the number of p_n reaches a certain number .That is, delete the sample u, update $p_{n-1} = p_n - \{u\}$ and calculate the sample $R^{n-1}, \alpha_p^{n-1}, \alpha_s^{n-1}, b^{n-1}$ with the minimum value.
Step 4: go to Step 2 until it satisfies the stop iteration condition ($min\left(g\left(x\right)\right) \geq \eta$).

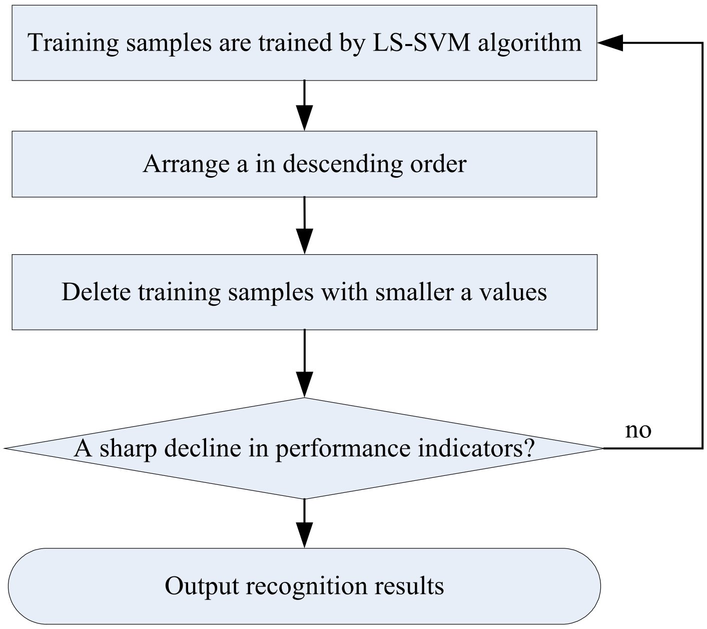
Training samples are trained by LS-SVM algorithm
Arrange a in descending order
Delete training samples with smaller a values
A sharp decline in performance indicators?
no
Output recognition results

Figure 2. Flowchart of sparsely LS-SVM algorithm

5 Application of Fast Sparse LS-SVM in Radar Target Recognition

5.1 Simulation Data

According to the above multiple scattering points model, four targets are chosen as test targets. "1" shaped targets, "V" shaped targets, "±" shaped targets and "÷" shaped targets. The number of scatter points for "1" targets is 5, and the scatter points of "V", "±" and "÷" are 9. In the experiment, the radar transmitting signal bandwidth is 150MHZ, and the distance resolution is 1m.

According to the formula:

$$x_n(m) = [|\, x_1(m)\,|,|\, x_2(m)\,|,\ldots,|\, x_n(m)\,|,]^{\mathrm{T}} \tag{41}$$

the high resolution range images of four kinds of simulation target "Hi", "V", "dry" and "small" in the range of 0° to 180° are calculated. The odd attitude angle samples are used as the learning samples, and the dual attitude angle samples are taken as the test samples, and the training sample set M and the test sample set X are obtained.

5.2 Simulation Analysis

5.2.1 Recognition Experiment Based on SVM

The experimental data are the distance image samples after the feature extraction of center distance and the feature extraction of invariant KPCA, and the SVM algorithm is used for the classification algorithm. In the experiment, the Gauss kernel function is used for the SVM, and the selection of the kernel function parameters and the penalty factor of the two planning has influence on the experimental results. Here, the cross validation algorithm is used to estimate the joint precision of parameters and each parameter in a certain range, so as to determine the best parameters of the actual problems and select the appropriate Experiment 1: SVM target recognition method for two target recognition performance based on translation invariant KPCA feature extraction in different attitude angles.

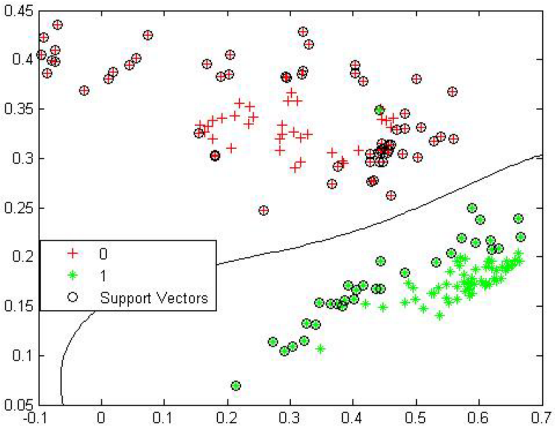

Figure 3. Results of the two class target recognition based on SVM

Table 1. Recognition results of two targets in different attitude angles by SVM method

target	correct recognition rate		
	0º-30º	0º-60º	0º-90º
“1”“V”	0.9947	0.9862	0.9815
“1”“±”	0.9824	0.9751	0.9613
“1”“÷”	0.9844	0.9712	0.9608
“V”“±”	0.9721	0.9634	0.9501
“V”“÷”	0.9708	0.9610	0.9535
“±”“÷”	0.9611	0.9497	0.9424
average recognition rate	0.9776	0.9678	0.9583

Experiment 2: SVM target recognition method to identify three targets in different attitude angles.

Table 2. Recognition results of three targets in different attitude angles by SVM method

target	correct recognition rate		
	0º-30º	0º-60º	0º-90º
“1”“V”“±”	0.8624	0.8301	0.8002
“1” “V”“÷”	0.8714	0.8234	0.7996
“1”“±”“÷”	0.8586	0.8297	0.8053
“V”“±”“÷”	0.8283	0.8014	0.7932
average recognition rate	0.8552	0.8212	0.7996

The noise of samples is 20dB, and the average of 200 experimental results is shown in tab.1 and tab.2. The data in the table show that the average recognition rate decreases with the increase of the attitude angle range, which is because with the increase of the attitude angle, the greater the difference between the target HRRP and the less distinguishable. Tab.1 compared with Tab.2, it is found that the recognition rate of the three class is lower than that of the two class target, which indicates that the three category is more difficult to classify.

5.2.2 Recognition Experiment Based on LS-SVM

In this experiment, the LS-SVM algorithm is used to classify and identify the one dimensional range profiles of four kinds of aircraft targets in the large attitude angle range, and the "One-against-one" multi class classification strategy is adopted. In this experiment, the Gauss radial basis kernel function is used in the experiment, and the selection of the kernel function parameter and the penalty factor is influenced by the selection of the kernel function parameter and the penalty factor gamma. The cross validation (Cross Validation) algorithm is selected to estimate the joint accuracy of each parameter in a certain range to determine the best parameters of the actual problem, and select the suitable LS-SVM. When the LS-SVM model. The LS-SVM algorithm is used to classify the one dimensional range profile of three radar targets. Through several experiments, it is found that when $\sigma^2 \in \lceil 0.1, 0.5 \rceil$ and $\gamma \in \lceil 10, 20 \rceil$, the experimental results are stable and the difference is not big and the recognition rate is high, so we take $\sigma^2 = 0.4$ and $\gamma = 10$ in this experiment.

Experiment 3: LS_SVM target recognition method for two target recognition performance based on translation invariant KPCA feature extraction in different attitude angles.

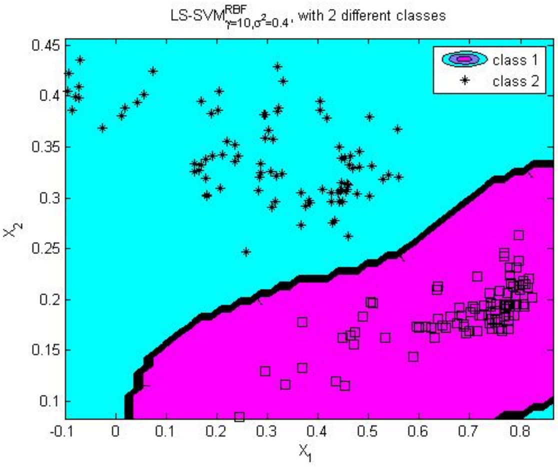

Figure 4. Results of two types of target recognition based on standard LS-SVM algorithm

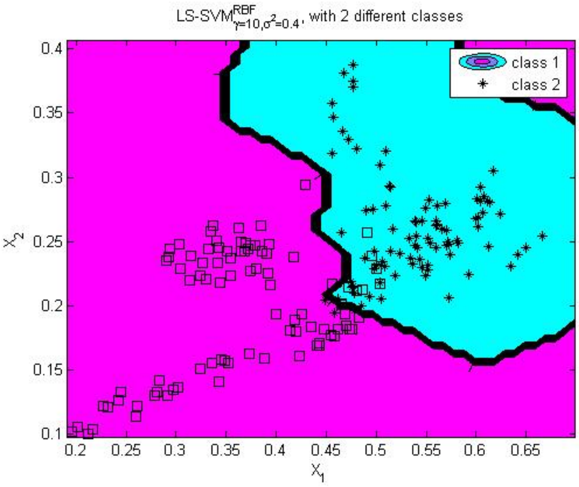

Figure 5. Results of two types of target recognition based on standard LS-SVM algorithm

Table 3. Recognition results of two targets in different attitude angles by LS_SVM method

target	correct recognition rate		
	0º-30º	0º-60º	0º-90º
“1”“V”	0.9926	0.9914	0.9897
“1”“±”	0.9905	0.9841	0.9812
“1”“÷”	0.9863	0.9789	0.9773
“V”“±”	0.9721	0.9762	0.9602
“V”“÷”	0.9731	0.9739	0.9611
“±”“÷”	0.9586	0.9578	0.9482
average recognition rate	0.9789	0.9771	0.9696

Experiment 4: LS_SVM target recognition method for three target recognition performance based on translation invariant KPCA feature extraction in different attitude angles.

Table 4. Recognition results of three targets in different attitude angles by LS_SVM method

target	correct recognition rate		
	0º-30º	0º-60º	0º-90º
“1”“V”“±”	0.8702	0.8648	0.8511
“1” “V”“÷”	0.8717	0.8613	0.8523
“1”“±”“÷”	0.8665	0.8542	0.8396
“V”“±”“÷”	0.8541	0.8519	0.8287
average recognition rate	0.8656	0.8581	0.8429

Tab.3 and Tab.4 show the average of the 100 experimental results. The data in the table show that the average recognition rate decreases with the increase of the attitude angle range, which is because with the increase of the attitude angle, the greater the difference between the target HRRP and the less distinguishable. At the same time, it can be found that using the LS-SVM algorithm, the three aircraft targets in each attitude angle range have a very high recognition rate, which reflects the superior recognition performance of the least squares support vector machine in the case of the non separable sample set.

Comparing the experimental results of SVM algorithm in tab.1 and LS-SVM algorithm in tab.3, it can be found that, under the same data, the recognition rate of LS-SVM is higher than that of the SVM algorithm, which shows that the LS-SVM algorithm is superior to the SVM algorithm in the recognition of radar range profile in the large attitude angle range. Perhaps in some pattern recognition areas, the LS-SVM algorithm is indeed better than the SVM algorithm, but this is at the expense of the support vector (in the LS-SVM algorithm, almost all training samples become support vectors).

5.2.3 Experiments Based on Improved LS-SVM Algorithm

In view of the lack of sparsity of the solution of LS_SVM, a sparse LS-SVM algorithm is given in this paper to further improve the recognition effect of LS-SVM algorithm in radar target recognition. In this experiment, we use the fast sparse LS-SVM algorithm proposed in this chapter to identify two types and three types of targets.

Experiment 5: fast sparse LS-SVM algorithm is used to identify two different targets. The results of the experiment are shown in tab.5.

Table 5. Fast sparse LS_SVM method for two target recognition results

target	Sparse front sample number	Sparse post sample number	Training time /s	Classification accuracy
“1”“V”	60	20	3	0.8933
“1”“±”	60	18	2	0.9027
“1”“÷”	60	19	3	0.8945
“V”“±”	60	23	4	0.8526
“V”“÷”	60	26	5	0.8415
“±”“÷”	60	25	4	0.8639

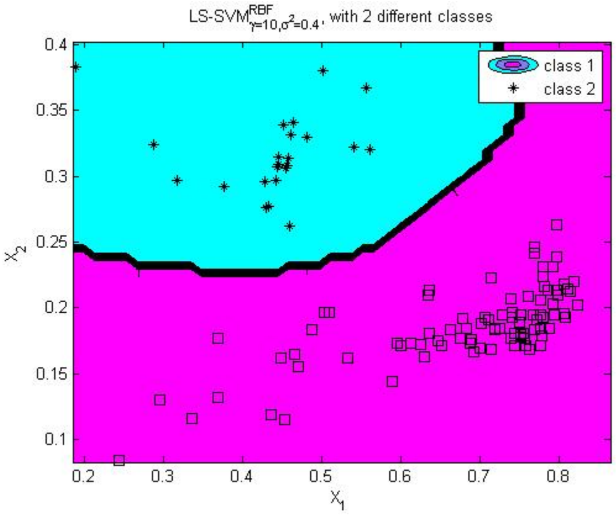

Figure 6. Results of two kinds of target recognition based on fast sparse LS-SVM algorithm

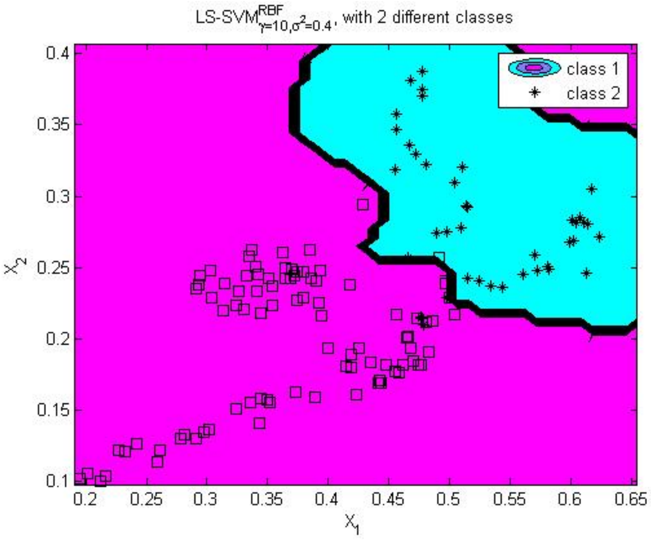

Figure 7. Results of two kinds of target recognition based on fast sparse LS-SVM algorithm

Experiment 6: fast sparse LS-SVM algorithm is used to identify three different targets. The results of the experiment are shown in tab.6.

Table 6. Fast sparse LS_SVM method for three target recognition results

target	sparse front sample number	sparse post sample number	training time /s	classification accuracy
“1”“V”“±”	60	30	3	0.8651
“1” “V”“÷”	60	28	4	0.8429
“1”“±”“÷”	60	31	4	0.8468
“V”“±”“÷”	60	33	5	0.8317

Tab.5 and tab.6 are the classification results of the fast sparse LS-SVM algorithm, and it is obvious that the number of training samples is greatly reduced after reasonable sparsity with Fig.4 and fig.5. At this time, if an unknown sample is classified, it not only reduces the calculation of the kernel function, but also reduces the number of summation, and improves the efficiency of recognition. At the same time, using the training sample set after sparsity and the Iterative Incremental LS-SVM algorithm of the 4.3 section, it can easily calculate the inverse of the kernel correlation matrix, which not only ensures that the classification recognition rate is basically kept the same, but also reduces the iteration number of the Iterative Incremental LS-SVM algorithm and avoids the mass matrix. The inverse problem is beneficial to the implementation of hardware. Experiments show that the new fast sparse LS-SVM algorithm proposed in this chapter has great advantages.

5.2.4 Comparison of the Three Methods

Experiment 7: SVM, LS-SVM and fast sparse LS-SVM algorithm for recognition of two kinds of targets.

Select the two types of "±" "V" to identify the target, use the three classification algorithms involved in this chapter: SVM, LS-SVM and fast sparse LS-SVM algorithm, the experimental results are shown in tab.7.

Table 7. Recognition results of two targets by SVM, LS-SVM and fast sparse LS-SVM algorithm.

algorithm	support vector number	training time	correct recognition rate
SVM	40	3	0.8643
LSSVM	60	4	0.9046
sparse LS-SVM	30	2	0.9021

Observation tab.7 shows that for all values, the classification accuracy of fast sparse LS-SVM algorithm is better than LS-SVM approximation than SVM. The support vector number of the fast sparse LS-SVM algorithm is 0.5 times that of LS-SVM, and the number of support vectors is obviously smaller than the support vector machine SVM, so the training time of the sparse LS-SVM is 0.5 times that of the LS-SVM training time. The fast sparse LS-SVM algorithm proposed in this chapter has obvious advantages, so it can be concluded that the fast sparse LS-SVM algorithm is the best choice for high resolution distance image.

Experiment 8: recognition of three types of recognition methods under different feature extraction methods

Table 8. Recognition results of three types of recognition methods under different feature extraction methods

algorithm	SVM	LS-SVM	Fast sparse LS-SVM
center distance feature extraction	0.8621	0.8764	0.8783
translation invariant KPCA feature extraction	0.8934	0.9136	0.9162

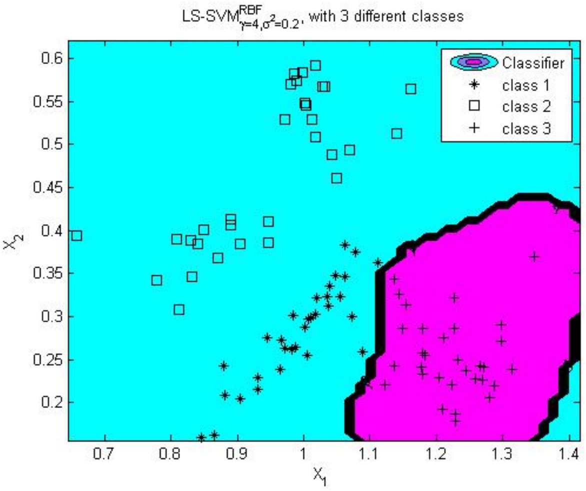

Figure 8. Results of three types of target recognition based on fast sparse LS-SVM algorithm with the feature extraction of zero shift translation invariant

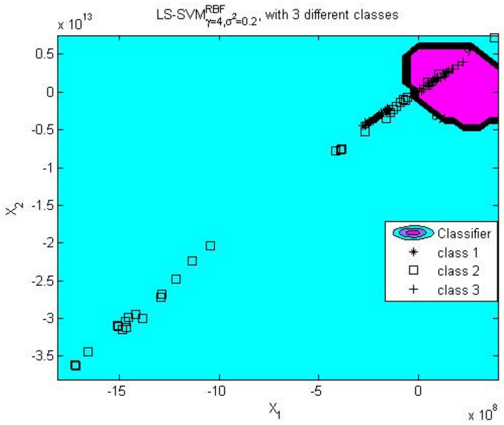

Figure 9. Results of three types of target recognition based on fast sparse LS-SVM algorithm with the feature extraction of center distance

From the tab.8, we can see that the method of feature extraction has a great influence on the classification performance. The recognition rate of the three classifiers in the feature extraction of the center distance feature and the translation invariant KPCA is very different. From the data, the result of recognition and classification after the translation of the translation invariant KPCA feature is obviously better than the result of the center distance feature extraction. From the comparison of classification results in Fig.8 and Fig.9, it is obvious that these two methods can affect the classifier performance.

6 Summary

In this paper, the principle of support vector machine (SVM) is described in detail. On this basis, the least squares support vector machine (LS-SVM) algorithm is introduced. In view of the lack of sparsity of LS-SVM training samples, a fast sparse least squares support vector machine (FSALS-SVM) algorithm is proposed, followed by SVM, LS-SVM and fast sparse LS-SVM three. The algorithm is applied to the recognition of radar HRRP. Through the experiment of classification and recognition of one dimensional range image data of radar target, the fast sparse LS-SVM algorithm can make the classifier more sparse on the premise of obtaining the same generalization performance.

References

1. Ebuchi F, Kitamura T. Fast Sparse Least Squares Support Vector Machines by Block Addition[C]. International Symposium on Neural Networks. Springer, Cham, 2017:60-70.
2. Shao Y H, Li C N, Liu M Z, et al. Sparse L q -norm least squares support vector machine with feature selection[J]. Pattern Recognition, 2018, 78:167-181.
3. Ma W, Liu H. Sparse least square support vector machines based on random entropy[C].Control Conference. IEEE, 2017:333-337.
4. Zhang C H, Chen Y, Cao S H, et al. A Sparse Algorithm Based on Tsallis Entropy for Least Squares Support Vector Machine[J]. Fire Control & Command Control, 2017.
5. Yang Z, Zhou Z, Jiang Y. Least squares support vector machine with parametric margin for binary classification1[J]. Journal of Intelligent & Fuzzy Systems, 2016, 30(5):2897-2904.
6. Liu D, Shi Y, Tian Y, et al. Ramp loss least squares support vector machine[J]. Journal of Computational Science, 2016, 14:61-68.
7. Si G, Shi J, Zhang G, et al. Density clustering pruning method based on reconstructed support vectors for sparse LSSVM[C].Control and Decision Conference. IEEE, 2016:3582-3587.
8. Peng J X, Rafferty K, Ferguson S. A fast algorithm for sparse support vector machines for mobile computing applications[J]. Neurocomputing, 2016, 230.

9. Nan S, Sun L, Chen B, et al. Density-Dependent Quantized Least Squares Support Vector Machine for Large Data Sets.[J]. IEEE Transactions on Neural Networks & Learning Systems, 2017, 28(1):94-106.
10. Shi J, Si G, Guo Z, et al. A pruning strategy based on confidence interval for sparse LS-SVM[C].World Congress on Intelligent Control and Automation. 2016:577-582.
11. Wong P K, Hang C W, Chi M V, et al. Online wavelet least-squares support vector machine fuzzy predictive control for engine lambda regulation[J]. International Journal of Engine Research, 2016, 17(8).
12. Mageswari A U, Vinodha R. Engine knock detection based on wavelet packet transform and sparse fuzzy least squares support vector machines (SFLS-SVM)[J]. Iioab Journal, 2016, 7(11):194-199.
13. Yang Z, Zhou Z, Jiang Y. Least squares support vector machine with parametric margin for binary classification1[J]. Journal of Intelligent & Fuzzy Systems, 2016, 30(5):2897-2904.