RESEARCH PAPER | Mathematics | Volume : 4 | Issue : 4 | Apr 2014 | ISSN - 2249-555X

Learning Algorithm of Svm Reduce The Optimization Error And Give The Maximum Accuracy of The QP

KEYWORDS: SVM Classification, Soft Margin, Optimal Criteria, Direct Iterative Process.

M.Premalatha
Research Scholar, Department of Mathematics Sathyabama University, Chennai, T.N, India.

Dr.C.Vijayalakshmi
School of Advance Science, Department of mathematics, VIT University, Chennai, T.N, India.

ABSTRACT *The field of machine learning is concerned with constructing computer program that automatically improve its performance with experience. SVMs (Support Vector Machines) are a useful technique for data classification. Support Vector Machine (SVM) is a linear machine working in the highly dimensional feature space formed by the nonlinear mapping of the N-dimensional input vector x into a K-dimensional feature space (K>N) through the use of a mapping Φ (x). The data points corresponding to the non-zero weights are called support vectors. The main goal is to measure the error to get the exact solution can be approximated by a function and also get the error accurately to determine the best function implemented by learning system using finite training set and testing set (unseen). The best function closely measure the optimization error in finite training set then the function have less approximation to lead a large estimation error. The main goal of learning algorithm is minimize the training set or time. Smaller constraint by the number of training data, the error is dominated by the approximation then the optimization error can be reduced the iterative time.*

INTRODUCTION

Machine learning system is trained by using a sample set of training data. SVMs estimate a linear decision function; mapping of the data into a higher-dimensional feature space may be needed. This mapping is characterized by the choice of a class of functions known as kernels [1]. The foundations of Support Vector Machines (SVM) have been developed by Vapnik [2]. A step in SVM classification involves identification as which are intimately connected to the known classes. This is called feature selection or feature extraction. Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data. Support Vector machines can be defined as systems which use hypothesis space of a linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. Each instance in the training set contains one target values and several variables.

SVM CLASSIFICATION

The training set is said to be linearly separable when there exists a linear discriminant function whose sign matches the class of all training examples. When a training set is linearly separable there usually is infinity of separating hyperplane. When the data set is large this optimization problem becomes very challenging, because the quadratic form is completely dense and the memory requirements grow with the square of the number of data points. We present a decomposition algorithm that guarantees global optimality, and can be used to train SVM's over very large data sets (1, 00,000 data points) [3]. The main idea behind the decomposition is the iterative solution of sub-problems and the evaluation of optimality conditions which are used both to generate improved iterative values, and also establish the stopping criteria for the algorithm.

Optimal Hyperplane

The SVM classification technique and show how it leads to the formulation of a QP programming problem in a number of variables that is equal to the number of data points. The data set is linearly separable, and to find the best hyperplane that separates the data [4].

$$\min(w,b) = \frac{1}{2}\|w\|^2 \qquad (1)$$
$$y_i(w^T\phi(x_i)+b \geq 1$$

$$f_{w,b} = \frac{sign(w.x+b)}{\|w\|} \leq A$$

Dual problem:

$$\max D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i\alpha_i y_j\alpha_j\phi(x_i)^T\phi(x_j)$$
$$\alpha_i \geq 0$$
$$\sum_i y_i\alpha_i = 0 \qquad (2)$$

The linear discriminant Function

$$\vec{y} = \sum_{i=1} y_i\alpha_i^*\phi(x_i)^T\phi(x) + b^* \qquad (3)$$

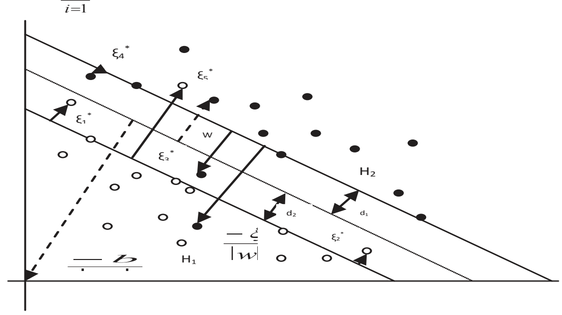

Fig: 1 Linear and Non Linear Separable

Soft Margin Hyperplane

The dual formulation of this soft-margin problem is strikingly similar to the dual formulation (2) of the optimal hyperplane algorithm. The only change is the appearance of the upper bound C for the coefficients α.

$$\min(w,b,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad (4)$$
$$y_i(w^T\phi(x_i) + b \geq 1 - \xi_i$$

Ξ is a slack variables C is the additional parameter that controls the compromise between the large margin and small margin.

$$\max D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i\alpha_i y_j\alpha_j K(x_i, x_j)$$
$$0 \leq \alpha_i \leq C$$
$$\sum_i y_i\alpha_i = 0 \quad (5)$$

Soft-Margin SVM problem (4) using the standard dual formulation (5), after computing the solution α*, the SVM discriminant function is

$$\vec{y} = \sum_{i=1} y_i\alpha_i^* K(x_i, x) + b^* \quad (6)$$

The box constraints $A_i \leq \alpha_i \leq B_i$ and the equality constraint $\Sigma\alpha_i = 0$ define the feasible region, the domain of α values that satisfy the constraints. The optimal bias b_i can be determined by returning to the primal problem, the box constraint $0\leq \alpha_i \leq C$ as box constraint on the quantity $y_i\alpha_i$:

$$y_i\alpha_i \in [A_i, B_i] = (0, C) \quad if \quad y_i = +1$$
$$(-C, 0) \quad if \quad y_i = -1 \quad (7)$$

We can represent these constraints using positive Lagrange coefficients $\alpha_i \geq 0$.

$$L(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(w^T\phi(x_i) + b)$$
$$-1 + \xi_i \quad (8)$$
$$D(\alpha) = \min L(w)$$
$$\xi_i \geq 0$$

$$\vec{D} = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i\alpha_i y_j\alpha_j K(x_i, x_j)$$
$$if \sum y_i\alpha_i = 0 \quad ; \alpha_i \leq C$$
$$= -\infty \quad ; otherwise$$

The dual problem (5) is the maximization of this expression subject to positivity constraints $\alpha_i \geq 0$. The conditions $y_i\alpha_i = 0$ and $y_i \alpha_i \leq C$ appear as constraints in the dual problem because the cases where D (α) = minus infinity are not useful for a maximization.

$$D(\alpha) = \vec{D}(\alpha) \leq L(w) \leq P(w)$$
$$D(\alpha^*) = P(w^*) \quad (9)$$

Suppose we can find α* and (w*, b*, ξ*) such that D (α*) = P (w*, b*, ξ*). Convex optimization problems with linear constraints are known to have such solutions. This is called strong duality.

OPTIMALITY CRITERIA

Let α*= (α_1*, α_2*, α_3*… α_n*) be solution of the dual problem (5). Obviously α* satisfies the dual constraints. Let d*= (d_1*, d_2*, d_3*,…, d_n*) be the derivatives of the dual objective function in α*

$$d_i^* = \frac{\partial D(\alpha^*)}{\partial \alpha_i} = 1 - y_i\sum_{j=1}^{n} y_j\alpha_j^* K(x_i, x_j)$$
$$y_i\alpha_i^* < B_i \quad A_j < y_j\alpha_j^* \quad (10)$$

$$\alpha_k^\varepsilon = \alpha_k^* = \begin{pmatrix} +\varepsilon y_k & if & k = i \\ -\varepsilon y_k & if & k = j \\ 0 & otherwise & \end{pmatrix}$$
$$D(\alpha^\varepsilon) - D(\alpha^*) = \varepsilon(y_i d_i^* - y_j d_j^*) + o(\varepsilon)$$
$$y_i d_i^* - y_j d_j^* \quad is \quad negative$$

$$\max_{i\in Iup} y_i d_i^* \leq \beta \leq \min_{i\in Idown} y_j d_j^*$$
$$zI_{up} = y_i\alpha_i < B_i \quad ; I_{down} = y_j\alpha_j > A_j$$
$$if \quad y_k d_k^* > \beta \quad then \quad y_k\alpha_k^* = B_k$$
$$if \quad y_k d_k^* < \beta \quad then \quad y_k\alpha_k^* = A_k$$
$$if \quad d_k^* > y_k\beta \quad then \quad \alpha_k^* = C$$
$$if \quad d_k^* < y_k\beta \quad then \quad \alpha_k^* = 0$$
$$w* = \sum y_k\alpha_k^*\phi(x_k), \quad b^* = \beta \quad \xi_k^* = \max(0, d_k^* - y_k\beta)$$

These values satisfy the constraints of the primal problem (4).

Support Vectors

A short derivation using (10) then gives

$$P(w^*) - D(\alpha^*) = C\sum_{k=1}^{n}\xi_k^* - \sum_{k=1}^{n}\alpha_k^* d_k^* = \sum_{k=1}^{n}(C\xi_k^* - \alpha_k^* d_k^*)$$

$$(C\xi_k^* - \alpha_k^* d_k^*) = -y_k \alpha_k^* \beta$$

$$d_k^* \le \quad or \quad \ge \quad y_k \beta$$

$$P(w^*) - D(\alpha^*) = -\beta\sum_{i=1}^{n} y_i \alpha_i^* = 0$$

$$D(\alpha^*) = P(w^*)$$

Support Vectors

$$d_k - y_k\beta = 1 - y_k \sum y_i \alpha_i K_{ik} - y_k b^* = 1 - y_k \vec{y}(x_k)$$

$$if \quad y_k \vec{y}(x_k) < 1 \quad then \quad \alpha_k = C \quad bounded \quad \sup port \quad vectors$$

$$if \quad y_k \vec{y}(x_k) > 1 \quad then \quad \alpha_k = 0 \quad not \quad \sup port \quad vectors$$

$$if \quad y_k \vec{y}(x_k) = 1 \quad 0 < \alpha_k < C \quad free \quad \sup port \quad vectors$$

Let B represent the best error achievable by a linear decision boundary in the chosen feature space. When the training set size n becomes large, one can expect about B_n misclassified training examples, that is to say $y_k \vec{y}(x_k) < 0$. All these misclassified data's are bounded support vectors [5] [6]. Therefore the number of bounded support vectors scales at least linearly with the number of data's. The total number of support vectors is asymptotically equivalent to $2B_n$.

SVM Linear mapping function

SVMs is to make use of a (nonlinear) mapping function Φ that transforms data in input space to data in feature space in such a way mapped back into input space via Φ^{-1}.

Linear Separable - when Φ is trivial

Positively labelled data points in R^2

$$\left\{\begin{pmatrix}5\\1\end{pmatrix},\begin{pmatrix}5\\-1\end{pmatrix},\begin{pmatrix}8\\1\end{pmatrix},\begin{pmatrix}8\\-1\end{pmatrix}\right\}$$

$$\left\langle\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix}\right\rangle$$

SVM that accurately discriminates the two classes. Since the data is linearly separable [7].

$$S_1 = \begin{pmatrix}1\\0\end{pmatrix} \quad , S_2 = \begin{pmatrix}5\\1\end{pmatrix} \quad , S_3 = \begin{pmatrix}5\\-1\end{pmatrix}$$

$$S_1 = (10) \quad \vec{S} = (101)$$

$$x_1\phi(S_1)\bullet\phi(S_1) + x_2\phi(S_2)\bullet\phi(S_1) + x_3\phi(S_3)\bullet\phi(S_1) = -1$$

$$x_1\phi(S_1)\bullet\phi(S_2) + x_2\phi(S_2)\bullet\phi(S_2) + x_3\phi(S_3)\bullet\phi(S_2) = 1$$

$$x_1\phi(S_1)\bullet\phi(S_3) + x_2\phi(S_2)\bullet\phi(S_3) + x_3\phi(S_3)\bullet\phi(S_3) = 1$$

$$x_1 \vec{S_1}\bullet\vec{S_1} + x_2 \vec{S_2}\bullet\vec{S_1} + x_3 \vec{S_3}\bullet\vec{S_1} = -1$$

$$x_1 \vec{S_1}\bullet\vec{S_2} + x_2 \vec{S_2}\bullet\vec{S_2} + x_3 \vec{S_3}\bullet\vec{S_2} = 1$$

$$x_1 \vec{S_1}\bullet\vec{S_3} + x_2 \vec{S_2}\bullet\vec{S_3} + x_3 \vec{S_3}\bullet\vec{S_3} = 1$$

The dot products results in

$$2x_1 + 6x_2 + 6x_3 = -1$$

$$6x_1 + 27x_2 + 25x_3 = 1$$

$$6x_1 + 25x_2 + 27x_3 = 1$$

$$x_1 = -2, x_2 = x_3 = 0.25$$

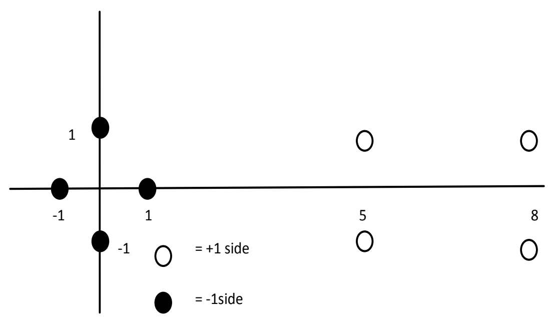

Fig: 2 Positively Labeled Data Points in R^2

$$\vec{w} = \sum_i x_i \vec{S_i}$$

$$= -2\begin{pmatrix}1\\0\\1\end{pmatrix} + 0.25\begin{pmatrix}5\\1\\1\end{pmatrix} + 0.25\begin{pmatrix}5\\-1\\1\end{pmatrix} = \begin{pmatrix}0.5\\0\\-1.5\end{pmatrix}$$

$$w = \begin{pmatrix}0.5\\0\end{pmatrix} and \quad b = -1.5$$

Given x, the classification f(x) is given by the equation where β (z)) returns the sign of z. classify the point x= (5, 6)

Volume : 4 | Issue : 4 | Apr 2014 | ISSN - 2249-555X

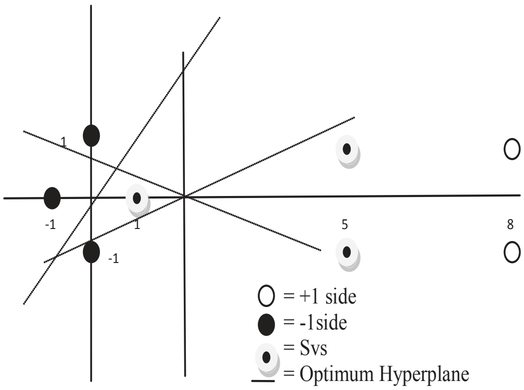

Fig: 3 linearly labeled data

$$f(x)=\beta\left(\sum_i x_i\phi(S_i)\bullet\phi(x)\right)$$

$$f\begin{pmatrix}5\\6\end{pmatrix}=\beta\left(-7\phi_1\begin{pmatrix}1\\1\end{pmatrix}\bullet\phi_1\begin{pmatrix}5\\6\end{pmatrix}+4\phi_1\begin{pmatrix}2\\2\end{pmatrix}\bullet\phi_1\begin{pmatrix}5\\6\end{pmatrix}\right)$$

$$\beta\left(-7\begin{pmatrix}1\\1\\1\end{pmatrix}\bullet\begin{pmatrix}0\\1\\1\end{pmatrix}+4\begin{pmatrix}2\\2\\1\end{pmatrix}\bullet\begin{pmatrix}0\\1\\1\end{pmatrix}\right)=\beta(-2)$$

DIRECT ITERATIVE PROCESS

Assume we are given a starting point x that satisfies the constraints of the quadratic optimization problem (5). A direction $v = (v_1 \ldots v_n)$ is a feasible direction if move the point α along direction v. The set S of all coefficients $\mu \geq 0$ such that the point $\alpha + \mu v$ satisfies the constraints [8] [9]. This set always contains 0; v is a feasible direction if S is not the singleton {0}. Because the feasible region is convex and bounded, the set S is a bounded interval of the form $[0, \mu_{max}]$. The simple optimization problem values of the D $(\alpha+\mu v)$ as a function of α.

$$\mu^* = \arg\max D(\alpha+\mu v) \qquad (11)$$

The location of its maximum α* is easily computed using Newton's formula

$$\mu^* = \frac{\dfrac{\partial D(\alpha+\mu v)}{\partial \mu}}{\dfrac{\partial^2 D(\alpha+\mu v)}{\partial^2 \mu}} = \frac{d^T v}{v^T v H}$$

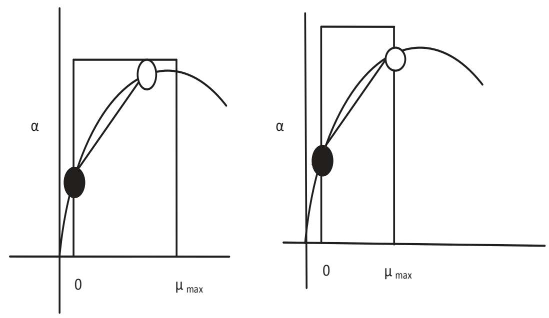

Fig: 4 feasible region μ

Where vector d and matrix H are the gradient and the Hessian of the dual objective function D (α),

$$d_i = 1 - y_i \sum y_j \alpha_j K_{ij}$$

$$\mu^* = \max(0, \min(\mu\max, \frac{d^T v}{v^T v H}))$$

This formula is the basis for a family of optimization algorithms. Starting from an initial feasible point, each iteration selects a suitable feasible direction and applies the direction Iterative formula (11) until reaching the maximum. The best direction v^{ij} requires iterating over the n (n − 1) possible pairs of indices.

$$v^* = \arg\max v \quad \max D(\alpha+\mu v^{ij}) - D(\alpha)$$
$$y_i\alpha_i + \mu \leq B_i$$
$$y_j\alpha_j - \mu \geq A_j$$

Maximal gain working set selection may reduce the number of iterations; it makes each iteration very slow. We may have to check the n (n − 1) possible pairs (i, j).

$$\max d^T v^{ij} = \max_{i\in I_{up}, j\in I_{down}} (y_i d_i - y_j d_j)$$
$$= \max y_i d_i - \min y_j d_j$$
$$i = \arg\max y_k d_k$$
$$j = \arg\min y_k d_k$$

This computation requires a time proportional to n.

Iterative Algorithm

Each iteration selects a working set and solves the corresponding sub problem using any suitable optimization algorithm [10] [11].

Iterative Algorithm:
Step: 1 Initial coefficient $\alpha_k \rightarrow 0$
Step: 2 Initial Iterative $d_k \rightarrow 1$
Step: 3 max $y_i d_i$; $y_i \alpha_i < B_i$
Step: 4 min $y_j d_j$: $A_j < y_j \alpha_j$
Step: 5 max ≤ min Optimality condition
Step: 6 Select a working set B contain 1 to n

Step: 7

$$\max \sum_{i=1}^{n} \alpha_i^t \left(1 - y_i \sum_{i,j \notin B}^{n} y_j \alpha_j K_{ij} \right)$$

$$-\frac{1}{2} \sum_i \sum_j y_i \alpha_i^t y_j \alpha_j^t K(x_i, x_j)$$

$$\sum_i y_i \alpha_i^t = -\sum_j y_j \alpha_j$$

Step: 8 Update iterative

$$d_k \rightarrow d_k - y_k \sum_{i=1}^{n} y_i (\alpha_i^t - \alpha_i) K_{ik}$$

Step: 9 update coefficient $\alpha_i' \rightarrow \alpha_i$

Numerical Accuracy

Numerical accuracy matters because many parts of the algorithm distinguish the variables α_i that has reached their bounds from the other variables. To solve the SVM dual optimization problem with accuracy that comfortably exceeds the needs of most machine learning applications. Approximate optimization can yield considerable speedups because there is no point in achieving a small optimization error when the estimation and approximation errors are relatively large [12 [13].

CONCLUSION

Once the system has learned, it is used to perform the required function based on the learning experienced. SVM learning algorithm is quickly reduce the optimization error comfortably below the expected approximation and estimation errors. Approximate optimization can yield considerable speedups because there is no point in achieving a small optimization error when the estimation and approximation errors are relatively large. In the case of Support Vector Machines, it remains difficult to achieve the benefits of these methods without partly losing the benefits of sparse solution. The iterative solution of sub-problems and the evaluation of optimality conditions which are used to generate improved iterative values reduce the optimization error and give the maximum accuracy of the QP finite training set and the optimization error can be reduced the iterative time.

Acknowledgment

Special Thanks to my Guide Dr.C.Vijayalakshmi for his continuous support encouragement and guidance.

REFERENCE

[1] Bottou. Fast kernel classifiers with online and active learning. Journal of Machine Learning Research, 6:1579–1619, September 2005. | [2] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. Data Min. Knowl. | [3] M.Premalatha, C.Vijayalakshmi (2011), "Machine Learning Classification Methods with Comparison Performance of Pattern Recognition." in Proc 2nd National Conference on Emerging Trends in Information Techno logy and Communication Systems "NCET 2011" Associated by DRDO, Chennai, Tamil Nadu PP.no.148-157. | [4] Tibshirani, R. and Hastie, T. (2007). Margin trees for high-dimensional classification, Journal of Machine Learning Research 8: 637–652 | [5] Fan, J. and Fan, Y. (2008). High dimensional classification using features annealed independence rules, Annals of Statistics. | [6] Kukar M (2006) Quality assessment of individual classifications in machine learning and data mining. Knowl Inf System 9(3):364–384. | [7] Tobias Glasmachers and Christian Igel. Maximum-gain working set selection for SVMs. Journal of Machine Learning Research, 7:1437–1466, July 2006. | [8] Pai-Hsuen Chen, Rong-En Fan, and Chih-Jen Lin. A study on SMO-type decomposition methods for support vector machines. IEEE Transactions on Neural Networks, 17:893– 908, July 2006. | [9] Ronan Collobert, Fabian Sinz, Jason Weston, and Leon Bottou. Large scale transductive svms. Journal of Machine Learning Research, 7:1687–1712, September 2006. | [10] Banerjee, O., Ghaoui, L. E. and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data, Journal of Machine Learning Research 9: 485–516. | [11] Don Hush, Patrick Kelly, Clint Scovel, and Ingo Steinwart. QP algorithms with guaranteed accuracy and run time for support vector machines. Journal of Machine Learning Research, 7:733–769, 2006. | [12] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Very large SVM training using Core Vector Machines. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT'05). Society for Artificial Intelligence and Statistics, 2005. | [13] Norikazu Takahashi and Tetsuo Nishi. Rigorous proof of termination of SMO algorithm for support vector machines. IEEE Transactions on Neural Networks, 16(3):774–776, 2005. |