SV MACHINES FOR FUNCTION ESTIMATION COVERING QUADRATIC PROGRAMMING WITH LARGE DATA SETS USING FOR INTERIOR POINT ALGORITHM

M. Premalatha
Research scholar, Department of Mathematics,
Sathyabama University, Chennai, Tamil Nadu, India
premalatha.maths@yahoo.co.in

Dr. C. VijayaLakshmi
School of Advance Science, Department of Mathematics,
VIT University, Chennai, Tamil Nadu, India.
vijusesha2002@yahoo.co.in

Abstract

Support Vector Machines (SVM) algorithms combine the simplicity and computational efficiency of linear algorithms, such as the perception algorithm or ridge regression, with the flexibility of nonlinear systems, like neural networks, and rigor of statistical approaches, as regularization methods in multivariate statistics, these algorithms typically reduce the learning step to a convex optimization problem that can always be solved in polynomial time, avoiding the problem of local minima typical of neural networks, decision trees and other nonlinear approaches. The problem of regression is that of finding a function which approximates mapping from an input domain to the real numbers based on a training sample. The basic idea underlying Support Vector (SV) machines for function estimation using interior point algorithm covering both primal and dual optimization extended to SVM regression function with large data sets. Solving by a predictor–corrector method iteratively and the similar data's are fused together to get maximum accuracy of optimization solution of large data.

Keywords: **SVM** Margin, SVM Regression, Interior point Algorithm, Convergence and Feasibility of SVM.

1. Introduction

A learning machine, such as the SVM, can be modelled as a function class based on some parameters α. Different function classes can have different capacity in learning, which is represented by a parameter h known as the VC dimension. The VC dimension measures the maximum number of training examples where the function class can still be used to learn perfectly, by obtaining zero error rates on the training data, for any assignment of class labels on these points [1] [2]. It can be proven that the actual error on the future data is bounded by a sum of two terms. The first term is the training error, and the second term if proportional to the square root of the VC dimension h. Thus, if we can minimize h, we can minimize the future error, as long as we also minimize the training error. SVM can be easily extended to perform numerical calculations [3]. One approach is to break a large optimization problem into a series of smaller problems, where each problem only involves a couple of carefully chosen variables so that the optimization can be done efficiently. The process iterates until all the decomposed optimization problems are solved successfully.

2. SVM Margin

Given some data $(\mathbf{x}_1 \ldots \mathbf{x}_i)$ and labels $(y_1 \ldots y_i)$, the SVM unit vector $\mathbf{w}_i$ obtained from this data is the centre of the largest hyper sphere that can fit inside the current space S_i. The position of $\mathbf{w}_i$ in the space S_i . Now, we can test each of the unlabeled instances $\mathbf{x}$ in the region to see how close their corresponding hyperplane in W come to the centrally placed $\mathbf{w}_i$. The closer a hyperplane in W is to the point $\mathbf{w}_i$, the more centrally it is placed in the space, and the more it bisects the space. For each unlabeled instance $\mathbf{x}$, the shortest distance between its hyperplane in and the vector $\mathbf{w}_i$ is simply the distance between the feature vector $\Phi(\mathbf{x})$ and the hyperplane $\mathbf{w}_i$ in F which is easily computed by $|\mathbf{w}_i \cdot \Phi(\mathbf{x})|$. These results in the natural rule: learn an SVM on the existing labeled Data [4] [6].

2.1. MaxMin Margin

The MaxMin approximation is designed given some data *(*$\mathbf{x}_1 \ldots \mathbf{x}_i$*)* and labels *(*$y_1 \ldots y_i$*)*, the SVM unit vector $\mathbf{w}_i$ is the centre of the largest hyper sphere that can fit inside the current space S_i and the radius m_i of the hyper sphere is proportional to the size of the margin of $\mathbf{w}_i$. We can estimate the relative size of the resulting version space *S by* labeling **x** as −1, finding the SVM obtained from adding **x** to our labeled training data of the margin -1. We can perform a similar calculation for *S*+ by relabeling **x** as class +1. An equal split of the space, *Area* (*S*−) and *Area* (*S*+) to be similar. Now, min (*Area* (*S*−), *Area* (*S*+)). It will be small if *Area* (*S*−) and *Area* (*S*+) are very different. Thus we will consider min (*-1, +1)* as an approximation and we will choose the **x** for which this quantity is largest. Hence, the MaxMin algorithm is as follows: for each unlabeled instance **x** compute the margins *-1* and *+1*, then choose the data unlabeled instance for which the quantity min (*-1, +1)* is greatest.

2.2. Ratio Margin

This method is similar in spirit to the MaxMin Margin method. We use *-1* and *+1* as indications of the sizes of *S*− and *S*+. The *relative* sizes of *-1* and *+1* are largest. The MaxMin and Ratio methods still hold even without the constraint on the modulus of the training feature vectors.

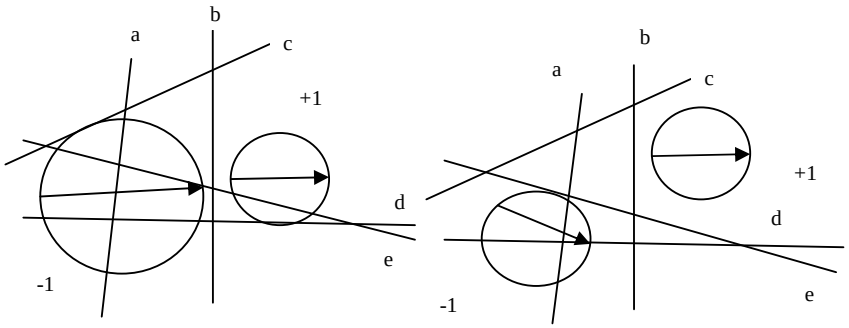

Fig.1 (a) MaxMin margin (b) Ratio Margin Primal Problem

3. Simplification of SVM

One drawback for using SVM in some real-life applications is the large number of arithmetic operations that are necessary to classify a new input vector. This number is proportional to the dimension of the input vector and the number of support vectors obtained. In the case of face detection, this is approximately 283,000 multiplications per pattern [8]. Since a closed form solution exists for the case of kernel functions that are 2nd. degree polynomials, we are using a simplified SVM in our current experimental face detection system that gains an acceleration factor of 20, without degrading the quality of the classifications [7].

3.1. Multiple Classifiers

The use of multiple classifiers offers possibilities that can be faster and/or more accurate have successfully combined the output from different neural networks by means of different schemes of arbitration in the face detection problem. Use a first classifier that is very fast as a way to quickly discard patterns that are clearly non-faces. This classification can be done about 300 times faster and is currently discarding more than 99% of input patterns [10].

3.2. SVM Regression

Let $x \in R^n$ and y ∈ **R,** where $\mathbf{R}^n$ represents input space. By some nonlinear mapping Φ, the vector *x* is mapped into a feature space in which a linear regression function is defined, $y = f(x, w) = w, \Phi(x) + b$. This *f* function based on independent uniformly distributed Data(x_1, y_1), (x_2,y_2),(x_3,y_3),….(x_m,y_m), the quadratic ∈-insensitive losses, with $\in = \theta - \gamma$

$$\min w = C\sum_{i=1}^{m} L^{\in}(x_i, y_i, f) + \frac{1}{2}\|w\|^2$$

Where *w* is weight vector and *c* is a constant parameter. Considering dual representation of a linear regression, *f* (*x*) can

$$f(x) = \sum_{i=1}^{m} \beta_i y_i K(x_i, x) + b = \sum_{i=1}^{m} \alpha_i K(x_i, x) + b$$

$$\max W(\alpha) = \sum_{i=1}^{m} y_i \alpha_i - \in \sum_{i=1}^{m} |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j (K(x_i, x_j) + \frac{1}{C} \delta_{ij}$$

$$\sum_{i=1}^{m} \alpha_i = 0$$

$$f(x) = \sum_{i=1}^{m} \alpha_i^* K(x_i, x) + b^*$$

where α_i^* is the solution of the quadratic optimization problem and b^* is chosen so

$$f(x_i) = y_i - \in - \frac{\alpha_i^*}{C}$$

For samples are inside the $\in$-tube, $x_i : (f(x_i)-y_i) < \in$, the corresponding α_i^* is zero.

$$f(x) = \sum_{i \in SV} \alpha_i^* K(x_i, x) + b^*$$

$$SV = i : |f(x) - y_i| \geq \in \quad or \quad \alpha_i^* > 0$$

α_i* are support vectors [9].

4. Interior Point Algorithm

An interior point algorithm is to compute the dual of the optimization problem and solve both primal and dual simultaneously. To iteratively find a feasible solution and to use the duality gap between primal and dual objective function to determine the quality of the current set of variables [5]. The objective function

$$\min \frac{1}{2} Q(x) + (c, x)$$

Subject to Ax= b $\quad s \leq x \leq t$ (1)

With c, x, s, t$\in$ R^n A$\in$ R^{n*m} b$\in$ R^m the inequalities between vectors holding component wise and $Q(x)$ being a convex function of x. Now we will add slack variables to all inequalities but the positivity constraints.

$$\min \frac{1}{2} Q(x) + (c, x)$$

Subject to Ax = b ; x – u = s , x – v = t ,s $\leq$ x $\leq$ t (2)

u, v $\geq$ 0 : x is free

With c, x, s, t $\in$ R^n A $\in$ R^{n*m} b $\in$ R^m

The dual of (2)

$$\max \frac{1}{2}(Q(x) - \partial Q(x), x)) + (b, y) + (s, z) - (t, p)$$

$$subject \ \ to \ \frac{1}{2} \partial Q(x) + c - (Ay)^T + p = z \tag{3}$$

p, z $\geq$ 0 : y is free

KKT Condition

$u_i z_i = 0$: $p_i v_i = 0$

A necessary condition of primal variables satisfies both optimality and feasibility condition of eq. (1) and eq.(2). It can be solved by Newton predictor–corrector method iteratively and the similar data's are fused together.

4.1. Different Parameters

If the parameters $C_{new} = \mu C_{old}$ is not optimized to use the *rescaled* values of the Lagrange multipliers as the new optimization problem solved by dual constraints.

The modified constraints x_{new} = μxold and likewise $b_{new} = \mu b_{old}$ (4)

In practice a speedup of approximately 95% of the overall training time can be observed when using the sequential minimization algorithm. The primal objective is convex, replace as μ^2.

4.2. SVM Convergence to Feasibility

In the case of both primal and dual feasible variables the following connection between primal and dual objective function holds:

Dual obj = primal obj – $\Sigma(u_i z_i + p_i v_i)$ (5)

In Regression Estimation (with the ε–insensitive loss function) one obtains for $\Sigma\ (u_i z_i + p_i v_i)$

$$\sum_i + \max(0,\ f(x_i) - (y_i + \varepsilon_i))(C - \alpha_i^*) - \min(0,\ f(x_i) - (y_i + \varepsilon_i))\alpha_i^*$$
$$+ \max(0, (y_i - \varepsilon_i^*) - f(x_i))(C - \alpha_i) - \min(0, (y_i - \varepsilon_i^*) - f(x_i))\alpha_i \quad (6)$$

To require

$$\frac{\sum u_i z_i + p_i v_i}{pri.obj + 1} \le \varepsilon \quad (7)$$

where convergence is measured in terms of the number of significant values in (7).

5. Solve by Interior Point Method

Solve a modified version thereof for some $\delta > 0$ substituted on the RHS in the first place and decrease δ while iterating. $u_i z_i = \delta \qquad p_i v_i = \delta$ (8)

Solving by a predictor–corrector approach until the duality space is small [12]. Solving non linear constraint in eq. (1) eq. (2) and eq. (8) accurately until to get the feasible solution.

$$A(x + \Delta x) = b$$
$$x + \Delta x - u - \Delta u = s$$
$$x + \Delta x + v + \Delta v = t$$
$$solve$$
$$A\Delta x = b - Ax = L$$
$$\Delta x - \Delta u = s - x - u = M$$
$$\Delta x + \Delta v = t - x - v = N$$
$$C + \frac{1}{2}\partial_x Q(x) + \frac{1}{2}\partial_x^2 Q(x)\Delta x - (A(y + \Delta y))^T + p + \Delta p = z + \Delta z$$
$$\frac{1}{2}\partial_x Q(x) = z + \Delta z - C - \frac{1}{2}\partial_x^2 Q(x)\Delta x + (A(y + \Delta y))^T - p - \Delta p$$
$$\frac{1}{2}\partial_x Q(x) = (A\Delta y)^T + \Delta z - \Delta p - C + (Ay)^T - p + z - \frac{1}{2}\partial_x^2 Q(x)\Delta x = \eta$$
$$(u_i + \Delta u_i)(z_i + \Delta z_i) = \delta$$
$$(p_i + \Delta p_i)(v_i + \Delta v_i) = \delta$$
$$u_i z_i + u_i \Delta z_i + z_i \Delta u_i + \Delta u_i \Delta z_i = \delta$$
$$u_i \Delta z_i + z_i \Delta u_i + \Delta u_i \Delta z_i = \delta - u_i z_i - \Delta u_i \Delta z_i$$
$$\Delta z_i + u_i^{-1} z_i \Delta u_i = \delta u_i^{-1} - (\Delta u_i \Delta z_i) u_i^{-1} - z_i = \beta_z$$
$$\Delta p_i + v_i^{-1} p_i \Delta v_i = \delta v_i^{-1} - (\Delta v_i \Delta p_i) v_i^{-1} - p_i = \beta_p$$

Where u_i^{-1} *denotes* the vector $(1/u_1 \ldots 1/u_n)$, and v_i^{-1}. Solving for Δu, Δv, Δz, Δp, we get

$$\Delta u_i = z_i^{-1} u_i (\beta_z - \Delta z_i); \quad \Delta z_i = u_i^{-1} z_i (m - \Delta x)$$
$$m = M - z_i^{-1} u_i \beta_z \tag{9}$$

$$\Delta v_i = p_i^{-1} v_i (\beta_s - \Delta p_i); \quad \Delta p_i = v_i^{-1} p_i (\Delta x - n)$$
$$n = N - p_i^{-1} v_i \beta_s \tag{10}$$

By KKT system

$$\begin{bmatrix} -H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \eta - u_i^{-1} zm - v_i^{-1} pn \\ L \end{bmatrix}$$
$$H = (\frac{1}{2} \partial_x^2 Q(x) + u_i^{-1} z + v_i^{-1} p \tag{11}$$

5.1. Iteration by Change of Choice

In the predictor step solve the group of constraints of (10) and (11) with $\delta = 0$ and all Δ–terms on the RHS set to 0, i.e. $\beta_z = z$, $\beta_p = p$. The values in Δ are back substituted in (10) and (11) are solved again in the corrector step and also estimate the corresponding values of x, p, v, z. Using the step length of the constraint of ξ.

$$\delta = \frac{(u,z) + (v,p)}{2n} \left(\frac{\xi - 1}{\xi + 10} \right)^2 \tag{12}$$

The eq. (12) is to use the average of the satisfaction of the KKT conditions (8); all variables (except *y*) are constrained to the solution to a feasible set

$$\begin{pmatrix} -\left(\frac{1}{2} \partial_x^2 Q(x) + 1 \right) & A^T \\ A & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix} \tag{13}$$

$$x = \max(x, \frac{g}{100})$$
$$u = \min(x - s, g)$$
$$v = \min(g - x, g) \tag{14}$$
$$z = \min(\Xi(\frac{1}{2} \partial_x Q(x) + C - (Ay)^T) + \frac{g}{100}, g)$$
$$p = \min(\Xi(-\frac{1}{2} \partial_x Q(x) + C - (Ay)^T) + \frac{g}{100}, g)$$

Ξ (.) denotes heavy side function. Ξ(x) =1; Ξ(x) =0 otherwise.

5.2. Further extended the interior point algorithm to SV Regression Estimation using predictor step

SV Regression $$Q(x) = \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i - x_j) + 2C \sum_{i=1}^{n} I(\alpha_i) + I(\alpha_i^*) \tag{15}$$

$$\partial_{\alpha 1} Q(\alpha) = \frac{\partial}{\partial \alpha_1} I(\alpha_i)$$
$$\partial^2_{\alpha i \alpha j} Q(\alpha) = K(x_i, x_j) + \mu_{ij} \frac{d^2}{d\alpha_i^2} I(\alpha_i) \tag{16}$$
$$\partial^2_{\alpha i \alpha_j^*} Q(\alpha) = -K(x_i, x_j)$$

and also $\partial^2_{\alpha_i^* \alpha_j^*} Q(\alpha), \partial^2_{\alpha_i^* \alpha_j} Q(\alpha)$

Matrix type $M = \begin{pmatrix} K + D & -K \\ -K & K + D^t \end{pmatrix}$

By applying an orthogonal transformation *M* can be inverted essentially by inverting an $n \times n$ matrix instead of a $2n \times 2n$ system. This to obtain the constant term *b* directly, therefore b = y.

6. Conclusion

Minimizing a constrained *primal* optimization problem one can ensure the maximizing dual optimization problem that the dual objective function optimality increases in each iteration. The minimum value of the objective function lies in the interval [dual objective *i*, primal objective *i*] for all steps *i*, hence every iteration increases the accuracy of the primal as well as dual constraint and satisfy the feasibility condition in each and every steps of the method to get the optimal solution of the both primal and dual and also duality of regression function of large data sets. The calculation of the primal objective function from the prediction errors is

$$\sum_{i.j} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K_{ij} = -\sum_i (\alpha_i - \alpha_i^*)(\phi + y_i + b)$$

If θ is a set of hyper parameters and α the dual variables, the learning of θ is to solve a min max problem the maximum of the dual is equal to the minimum of the primal: $\min_\theta \max_\alpha Dual(\alpha, \theta)$ by alternating between minimization on θ and maximization on α.

References

[1] Grossberg S. Adaptive Pattern Classification and Universal Recoding, I: Parallel Development and Coding of Neural Feature Detectors. Biological Cybernetics. 1976; 23 : 121-134

[2] M.Premalatha, C. Vijayalakshmi (2012), Analysis of Soft Computing in Neural Network, in Proc 2nd International Conference in Computer Applications'12, Associated with ASDF, ACM, SERSC, Pondicherry, Vol 2, PP.no.172-177

[3] Y. Q. Zhan and D. G. Shen, Pattern Recognition. 38, 157–161 (2005). Design Efficient Support Vector Machine for Fast Classification.

[4] Steinwart, I. (2002). Support vector machines are universally Consistent. *J. Complexity* 18 768-779.

[5] C.M. Bishop, Pattern Recognition and Machine Learning, *Information Science and Statistics*, Springer (2006)

[6] .Genov R, Cauwenberghs G, Keltron (2003): Support Vector "Machine" in Silicon IEEE *Trans. Neural Networks*, Vol. 14. No.5. pp.1426 -1433.

[7] Cristianini N, Shawe-Taylor J (2003): An introduction to Support Vector Machines and other kernel - based learning methods. *Cambridge, University Press*.

[8] Zhang L, Zhou W, Jiao L (2004): Wavelet Support Vector Machine. *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics*, Vol. 4. No.1. pp. 34 -39.

[9] Ji Zhu and Trevor Hastie. Kernel logistic regression and the import vector machine. Journal of Computational and Graphical Statistics, 14:185–205, 2005.

[10] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core vector machines: fast SVM training on very large datasets. Journal of Machine Learning Research, 6: 363–392, 2005.

[11] S. Sathiya Keerthi and Dennis M. DeCoste. A modified finite Newton method for fast solution of large scale linear SVMs. Journal of Machine Learning Research, 6:341–361, 2005.

[12] Stephen Boyd and Lieven Vandenberghe. Convex Optimization, Cambridge University Press, 2004.