Support Vector Regression for Outlier Removal

A dissertation submitted in partial fulfillment of the requirements for the M.Tech.(Computer Science) degree of Indian Statistical Institute

By

Gufran Malik
Roll No. CS0816

Under the supervision of

Prof C A Murthy
Machine Intelligence Unit

INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road
Kolkata-700108

Indian Statistical Institute

CERTIFICATE

This is to certify that the thesis entitled '*Support Vector Regression for Outliers Removal*' is submitted in the partial fulfillment of the degree of M.Tech in Computer Science at Indian Statistical Institute, Kolkata.

The work carried out by Mr. Gufran Malik under my supervision and guidance is adequate in scope and quality as a dissertation for the required degree.

Prof. CA Murthy
(Supervisor)

(External Examiner)

Contents

Acknowledgement

I take this opportunity to thank Prof CA Murthy, Machine Intelligence Unit, ISI Kolkata for his valuable guidance, inspiration. His pleasant and encouraging words have always kept me sprits up. I am grateful to him for providing me to work under his supervision. Also, I am very thankful to him for giving me the idea behind the algorithm developed in thesis work.

Finally, I would like to thank all my colleagues, class mates, friends, and my family members for their support and motivation.

Gufran Malik
M.Tech.(CS) 2008-10

Chapter 1. Introduction

1.1 Introduction to Problem

The Support Vector Machine (SVM) is a universal approach for solving the problems of multidimensional function estimation. Those approaches are all based on the Vapnik–Chervonenkis (VC) theory. Initially, it was designed to solve pattern recognition problems, where in order to find a decision rule with good generalization capability, a small subset of the training data, called the support vectors are selected. Experiments showed that it is easy to recognize high-dimensional identities using a small basis constructed from the selected support vectors. Recently, SVM has also been applied to various fields successfully such as classification, time prediction and regression. When SVM is employed to tackle the problems of function approximation and regression, the approaches are often referred to as the Support Vector Regression (SVR). The SVR type of function approximation is very effective, especially for the case of having a high-dimensional input space.

In general, for real-world applications, observations are always subject to noise or outliers. The intuitive definition of **outliers** is that "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". Outliers may occur due to various reasons, such as erroneous measurements or noisy phenomenon appearing in the tail portion of some noise distribution functions. However, the traditional SVR is not effective in dealing with outliers in training data commonly encountered in practical applications. Thus, even a few outliers result in a poor regression. The basic idea of the proposed method consists in gradually partitioning data into outliers and inliers, and thus refining the estimation with the inliers.

1.2 Brief Overview:

-Linear Regression

Regression analysis includes any techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. More specifically, regression analysis helps us understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables — that is, the average value of the dependent variable when the independent variables are held fixed.

Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships.

Regression models involve the following variables:

- The unknown parameters denoted as β; this may be a scalar or a vector of length *k*.
- The independent variables X.
- The dependent variable, *Y*.

A regression model relates *Y* to a function of X and β.

$$Y \approx f(\mathbf{X}, \boldsymbol{\beta})$$

In linear regression, the model specification is that the dependent variable, y_i is a linear combination of the *parameters* (but need not be linear in the *independent variables*).

Suppose we are given a data set $\{y_i,\, x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$of *n* statistical units, a linear regression model assumes that the relationship between the dependent variable y_i and the *p*-vector of regressors x_i is approximately linear. This approximate relationship is modeled through a so-called "disturbance term" ε_i — an unobserved random variable that adds noise to the linear relationship between the dependent variable and regressors. Thus the model takes form

$$y_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = x_i'\beta + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where ′ denotes the transpose, so that $x_i'\beta$ is the inner product between vectors x_i and β.

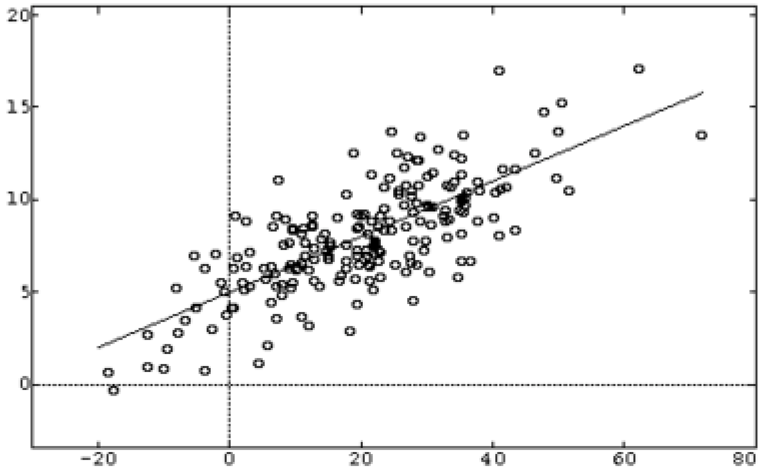
Often these *n* equations are stacked together and written in vector form as

$$y = X\beta + \varepsilon,$$

where

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Example of linear regression with one independent



variable.

-Support Vector Regression

Support vector machine (SVM) has been first introduced by Vapnik. There are two main categories for support vector machines: support vector classification (SVC) and support vector regression (SVR). SVM is a learning system using a high dimensional feature space. It yields prediction functions that are expanded on a subset of support vectors. A version of a SVM for regression has been proposed in 1997 by Vapnik, Steven Golowich, and Alex Smola . This method is called *support vector regression (SVR).* The model produced by support vector classification only depends on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR only depends on a subset of the training data, because the cost function for building the model ignores any training data that is close (within a threshold ε) to the model prediction.

Support Vector Regression (SVR) is the most common application form of SVMs. An overview of the basic ideas underlying support vector machines for regression and function estimation is also given in this paper.

In regression problems, we are given a training data set {(x_i; y_i)|i = 1,….., n} where $y_i \in R$ is called the observation and $x_i \in R^n$ is called the input data. These might be, for instance, exchange rates for some currency measured at subsequent days together with corresponding econometric indicators. The main goal of regression problems is to find a function f(x) that can correctly predict the observation values, y, of new input data points, x, by learning from the given training data set, S.

Here, learning from a given training data set means finding a linear or nonlinear surface that tolerates a small error in fitting this training data set.

In ε -SV regression, our goal is to find a function f(x) that has at most ε deviation from the actually obtained targets yi for all the training data, and at the same time is as flat as possible. In other words, we do not care about errors as long as they are less than ε , but will not accept any deviation larger than this. This may be important if you want to be sure not to lose more than ε money when dealing with exchange rates, for instance.

Also, applying the idea of support vector machines (SVMs) the function f(x) is made as flat as possible in fitting the training data. This problem is called ε -support vector regression (ε -SVR) and a data point $x_i \in R^n$ is called a **support vector** if $|f(x_i)-y_i| \geq \varepsilon$.

Conventionally, ε -SVR is formulated as a constrained minimization problem , namely, a convex quadratic programming problem or a linear programming problem . Such formulations introduce 2m more nonnegative variables and 2m inequality constraints that enlarge the problem size and could increase computational complexity for solving the problem.

For pedagogical reasons, we begin by describing the case of linear functions f, taking the form f(x)=<w,x>+b where <. , .> denotes the dot product in X. *Flatness* in the case of (1) means that one seeks a small w. One way to ensure this is to minimize the norm i.e. $\| w \|^2$=<w,w>. We can write this problem as a convex optimization problem.

minimize $\frac{1}{2} \| w \|^2$

subject to $y_i - \langle w, x_i \rangle - b \leq \varepsilon$;

$\langle w, x_i \rangle + b - y_i \leq \varepsilon$;

The tacit assumption was that such a function f actually exists that approximates all pairs (xi; yi) with ε precision, or in other words, that the convex optimization problem is *feasible*. Sometimes, however, this may not be the case, or we also may want to allow for some errors. Analogously to the ε soft margin. loss function which was adapted to SV machines, one can introduce slack variables ξ_i and ξ_i* to cope with otherwise infeasible constraints of the optimization problem . Hence we arrive at the formulation stated in

$$\min_{w,b,\xi_i,\xi_i*} \mathrm{R}(w,b\,,\,\xi_i,\xi_i*)=C\sum_{i=1}^{n}\ (\xi_i+\xi_i*) + \frac{1}{2}\| w \|^2$$

Subject to $y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i$;

$\langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i*$;

$\xi_i \geq 0$; $\xi_i* \geq 0$;

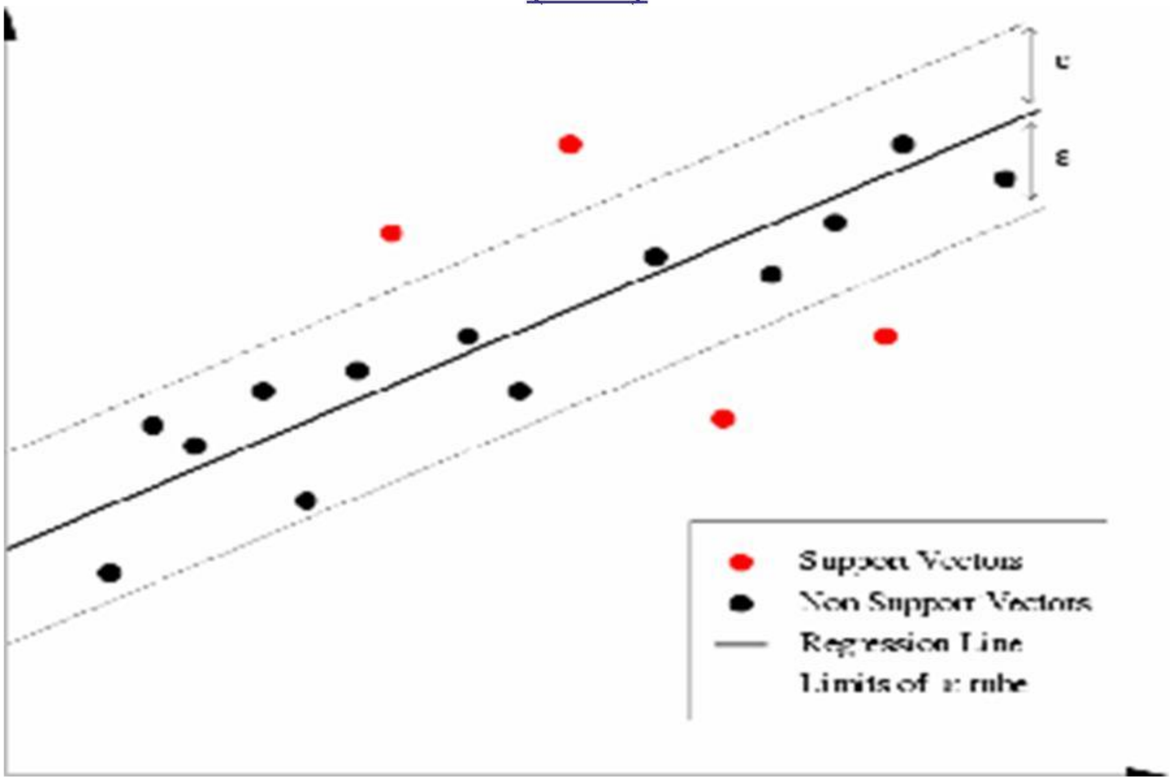
The constant C > 0 determines the trade-off between the flatness of f and the amount up to which deviations larger than ε are tolerated.

After solving this optimization problem one can get the function f(x)

as $$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i *) < x_i, x > +b$$

and this is the equation of hyper plane.

Principle of Support Vector Regression (SVR)



-Multivariate Regression Method:

The general purpose of multiple regression is to learn more about the relationship between several independent or predictor variables and a dependent or criterion variable. For example, a real estate agent might record for each listing the size of the house (in square feet), the number of bedrooms, the average income in the respective neighborhood according to census data, and a subjective rating of appeal of the house. Once this information has been compiled for various houses it would be interesting to see whether and how these measures relate to the price for which a house is sold. For example, you might learn that the number of bedrooms is a better predictor of the price for which a house sells in a particular neighborhood than how "pretty" the house is (subjective rating). You may also detect "outliers," that is, houses that should really sell for more, given their location and characteristics.

Regression is a generic term for all methods attempting to fit a model to observed data in order to *quantify the relationship* between two groups of variables. The fitted model may then be used either to merely *describe* the relationship between the two groups of variables, or to *predict* new values**.**

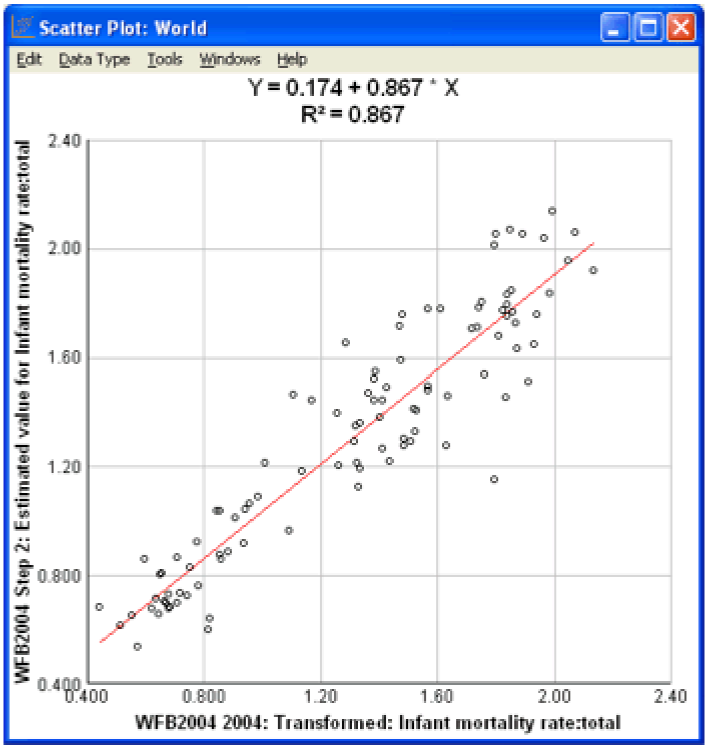
Personnel professionals customarily use multiple regression procedures to determine equitable compensation. You can determine a number of factors or dimensions such as "amount of responsibility" (*Resp*) or "number of people to supervise" (*No_Super*) that you believe to contribute to the value of a job. The personnel analyst then usually conducts a salary survey among comparable companies in the market, recording the salaries and respective characteristics (i.e., values on dimensions) for different positions. This information can be used in a multiple regression analysis to build a regression equation of the form:

Salary = .5*Resp + .8*No_Super

Once this so-called regression line has been determined, the analyst can now easily construct a graph of the expected (predicted) salaries and the actual salaries of job incumbents in his or her company. Thus, the analyst is able to determine which position is underpaid (below the regression line) or overpaid (above the regression line), or paid equitably.

In the social and natural sciences multiple regression procedures are very widely used in research. In general, multiple regression allows the researcher to ask (and hopefully answer) the general question "what is the best predictor of ...". For example, educational researchers might want to learn what are the best predictors of success in high-school. Psychologists may want to determine which personality variable best predicts social adjustment. Sociologists may want to find out which of the multiple social indicators best predict whether or not a new immigrant group will adapt and be absorbed into society.

The general computational problem that needs to be solved in multiple regression analysis is to fit a straight line to a number of points.



$Y = 0.174 + 0.867 * X$

$R^2 = 0.867$

An example of regression

In the more general multiple regression model, If there are n observation and *p* independent variables:

$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \varepsilon_i,$ for i=1,2,……,n

The least square parameter estimates are obtained by *p* normal equations. The residual can be written as

$e_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_1 - \cdots - \hat{\beta}_p x_p.$ for i=1,2,……,n

The **normal equations** are

$$\sum_{i=1}^{n}\sum_{k=1}^{p} X_{ij}X_{ik}\hat{\beta}_k = \sum_{i=1}^{n} X_{ij}y_i,\ j = 1, \ldots, p.$$

Note that for the normal equations depicted above, $y_i = \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \varepsilon_i$

That is, there is no β0. Thus in what follows, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p).$

In matrix notation, the normal equations for *k* responses (usually *k* = 1) are written as

$${}_{\mathbf{p}}(\mathbf{X}_{\mathbf{n}}^{\top}\mathbf{X})_{\mathbf{p}}\hat{\boldsymbol{\beta}}_{\mathbf{k}} ={}_{\mathbf{p}} \mathbf{X}_{\mathbf{n}}^{\top}\mathbf{Y}_{\mathbf{k}}.$$

with generalized inverse (–) solution, subscripts showing matrix dimensions:

$${}_{\mathbf{p}}\hat{\boldsymbol{\beta}}_{\mathbf{k}} = {}_{\mathbf{p}}(\mathbf{X}_{\mathbf{n}}^{\top}\mathbf{X})_{\mathbf{p}}^{-}\mathbf{X}_{\mathbf{n}}^{\top}\mathbf{Y}_{\mathbf{k}}.$$

Chapter 2.Support Vector Regression

Problem Statement:

In Support Vector Regression method , our goal is to find a function f(x) that has at most ε deviation from the actually obtained targets yi for all the training data, and at the same time is as flat as possible. In other words, we do not care about errors as long as they are less than ε , but will not accept any deviation larger than this.

Procedure:

As computationally powerful tools for supervised learning, support vector machines (SVMs) are widely used in classification and regression problems. Let us suppose that a data set $D = \{f(x_i; y_i)|i = 1,....., n\}$ is given for training, where the input vector $x_i \in R^d$ and y_i is the target value. SVMs take the idea to map these input vectors into a high dimensional, where a linear machine is constructed by minimizing a regularized functional. The linear machine takes the form of

f(x)=<w,x>+b, where , *b* is known as the bias, and <w,x> denotes the inner product.

The regularized functional is usually defined as

$$R(w,b) = C.\sum_{i=1}^{n} l(y_i, f(x_i)) + \frac{1}{2}\| w \|^2 \qquad (1)$$

Where the $C > 0$ is the regularization parameter and $\sum_{i=1}^{n} l(y_i, f(x_i))$ is empirical loss term. In standard SVMs, the regularized functional can be minimized by solving a convex quadratic programming optimization problem that guarantees a unique global minimum solution.

Various loss functions can be used in SVMs that result in quadratic programming. There are four popular loss functions widely used for regression problems. They are

1. Laplacian loss function: $l_1(\delta) = |\delta|$

2. Huber's loss function:
$$l_h(\delta) = \frac{\delta^2}{4\varepsilon}, \qquad \mathit{if}\ |\delta| \leq \varepsilon$$
$$l_h(\delta) = |\delta| - \varepsilon \qquad \text{o.w.}$$

3. ε -insensitive loss function:
$$l_\varepsilon(\delta) = 0\ , \quad \text{if } |\delta| \leq \varepsilon$$
$$l_\varepsilon(\delta) = |\delta| - \varepsilon \quad \text{o.w.}$$

4. Gaussian loss function: $l_g(\delta) = \frac{1}{2}\delta^2$

We will make use of ε -insensitive loss function for our problem.

We introduce ε -insensitive loss function into the regularized functional (1) that will leads to a quadratic programming problem that could work as a general framework. As usual, two slack variables are ξ_i and ξ_i * introduced as

$\xi_i \geq y_i - \langle w, x_i \rangle - b - \varepsilon$

$\xi_i{}^* \geq \langle w, x_i \rangle + b - y_i - \varepsilon$

The minimization of the regularized functional (1) with ε-insensitive loss function as loss function could be rewritten as the following equivalent optimization problem, which is usually called *primal* problem:

$$\min_{w,b,\xi_i,\xi_i*} \quad \mathrm{R}\,(\mathrm{w,b}\,,\ \xi_i\,,\xi_i\,*)=\mathrm{C}\sum_{i=1}^{n}\ (\xi_i+\xi_i\,*)\ +\ \frac{1}{2}\,\|\,w\,\|^2$$

Subject to $\quad y_i - \langle w,x_i\rangle - b \leq \varepsilon + \xi_i\ ;$

$$\langle w,x_i\rangle + b - y_i \leq \varepsilon + \xi_i\,*\ ; \qquad (2)$$

$$\xi_i \geq 0;\quad \xi_i\,* \geq 0;$$

The constant C>0 determine the trade-off between the flatness of f and amount up to deviations larger than ε are tolerated.

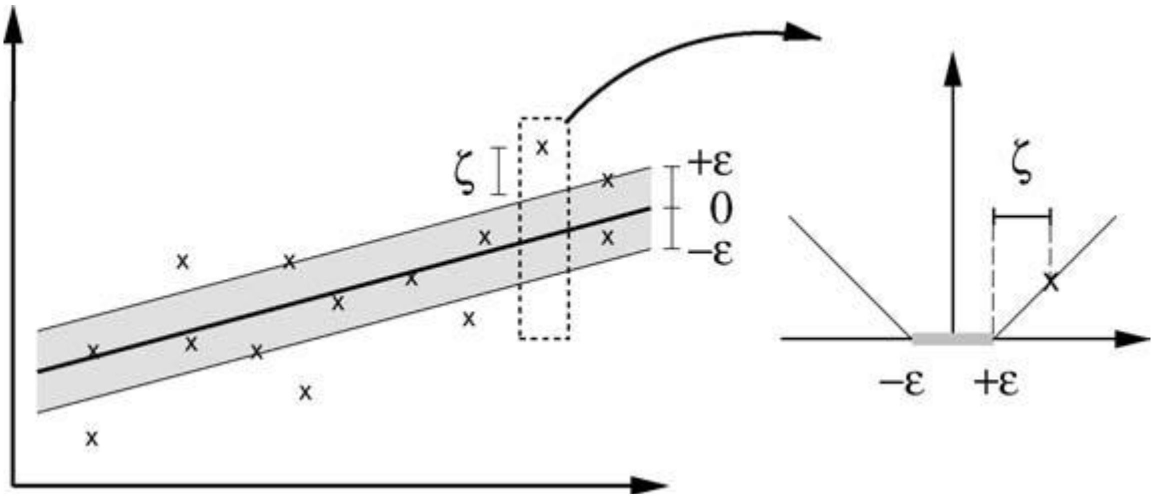


Fig. 1. *The soft margin loss setting for a linear SVM*

Figure 1 depicts the situation graphically. Only the points outside the shaded region contribute to the cost insofar, as the deviations are penalized in a linear fashion. It turns out that in most cases the optimization problem (2) can be solved more easily in its dual formulation.

Dual Problem and Quadratic:

The key idea is to construct a Lagrange function from the objective function (it will be called the *primal* objective function in the rest of this article) and the corresponding constraints, by introducing a dual set of variables. It can be shown that this function has a saddle point with respect to the primal and dual variables at the solution. For details see e.g. Mangasarian (1969), McCormick (1983) [22] and the explanations in appendix. We proceed as follows:

$$
\begin{aligned}
L = \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i *) - \sum_{i=1}^{n}(\eta_i\xi_i + \eta_i * \xi_i *) \\
- \sum_{\mathrm{i}=1}^{\mathrm{n}} \alpha_{\mathrm{i}}(\varepsilon + \xi_i - y_i + < w, x_i > + b) \\
- \sum_{\mathrm{i}=1}^{\mathrm{n}} \alpha_{\mathrm{i}} * (\varepsilon + \xi_i * + y_i - < w, x_i > - b)
\end{aligned}
\qquad (3)
$$

Here L is the Lagrangian and η_i , $\eta_i *$, α_{i} , $\alpha_{\mathrm{i}} *$ are Lagrange multipliers. Hence the dual variable in (3) has to satisfy positivity constraints, i.e.

$$\alpha_{\mathrm{i}} , \alpha_{\mathrm{i}} *, \ \eta_i \ , \eta_i * \geq 0 \qquad (4)$$

It follows from the saddle point condition that the partial derivatives of *L* with respect to the primal variables (w, b, ξ_i , $\xi_i *$) have to vanish for optimality i.e. The KKT conditions for the *primal* problem require

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} (\alpha_i * - \alpha_i) = 0$$

i.e. $$\sum_{i=1}^{n} (\alpha_i * - \alpha_i) = 0 \qquad (5)$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} (\alpha_i - \alpha_i *) x_i = 0$$

i.e. $$w = \sum_{i=1}^{n} (\alpha_i - \alpha_i *) x_i \qquad (6)$$

$$\frac{\partial L}{\partial \xi} = C - \alpha_i - \eta_i = 0$$

i.e. $$C = \alpha_i + \eta_i \qquad (7a)$$

$$\frac{\partial L}{\partial \xi *} = C - \alpha_i * - \eta_i * = 0$$

i.e. $$C = \alpha_i * + \eta_i * \qquad (7b)$$

Substituting the values from (5) ,(6) ,(7a) and (7b) into (3) yields the dual optimization problem.

$$\text{minimize} \quad \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i{*})(\alpha_j - \alpha_j{*}) < x_i + x_j > \\ -\varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i{*}) + \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i{*}) \end{cases} \qquad (8)$$

subject to $$\sum_{i=1}^{n}(\alpha_i - \alpha_i{*}) = 0$$

and $\alpha_i , \alpha_i{*}$ € [0,C]

In deriving (8) we already eliminated the dual variables $\eta_i , \eta_i{*}$ through equation (7a) and (7b) which can be reformulated as

$$\eta_i = C - \alpha_i$$

And $\eta_i{*} = C - \alpha_i{*}$

The *dual* problem (30) is also a constrained convex quadratic programming problem. Let us denote

$$\overset{\wedge}{\alpha} = [\alpha_1, \alpha_2, \ldots\ldots, \alpha_n, -\alpha_1{*}, \alpha_2{*}, \ldots\ldots\ldots, -\alpha_n{*}]^T$$

$$\hat{P} = [-y_1 + \varepsilon, -y_2 + \varepsilon, \ldots\ldots\ldots\ldots - y_n + \varepsilon, -y_1 - \varepsilon, -y_2 - \varepsilon, \ldots\ldots\ldots\ldots, -y_n - \varepsilon]^T$$

$$Q = \begin{bmatrix} K & -K \\ -K & K \end{bmatrix}$$

Where K is the *n by n* matrix whose entry is of the form $<x_i , x_j>$

Then equation (8) can be rewritten as

Min $\frac{1}{2}\hat{\alpha}^T Q\hat{\alpha}+P^T\hat{\alpha}$

$l_i \le \alpha_i \le u_i$ for all i .

and $\sum_{i=1}^{2n} \hat{\alpha_i} = 0$

$l_i = 0$, $u_i = C$ for $1 \le i \le n$

$l_i = -C$, $u_i = 0$ for $n+1 \le i \le 2n$

And this is a simple quadratic programming problem which can solved easily.

After solving this problem one can get w and b and then find the equation of hyper plane f(x) as described below.

$$w = \sum_{i=1}^{n}(\alpha_i - \alpha_i *)x_i$$

Thus $f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i *) < x_i, x > +b$

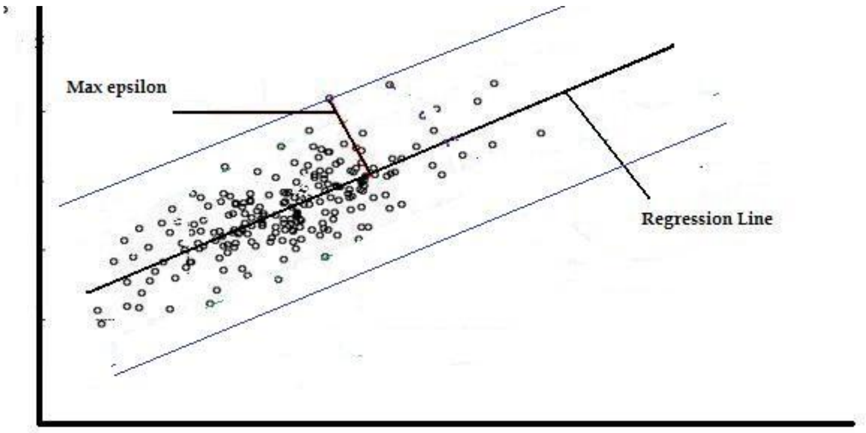
This is the so-called *Support Vector expansion*, i.e. *w* can be completely described as a linear combination of the training patterns x_i. In a sense, the complexity of a function's representation by SVs is independent of the dimensionality of the input space *X*, and depends only on the number of SVs. Moreover, note that the complete algorithm can be described in terms of dot products between the data. Even when evaluating *f* (*x*) we need not compute *w* explicitly. These observations will come in handy for the formulation of a nonlinear extension.

When there are no outlier present in the data set then usual Support Vector Regression give good results for the given value of epsilon.

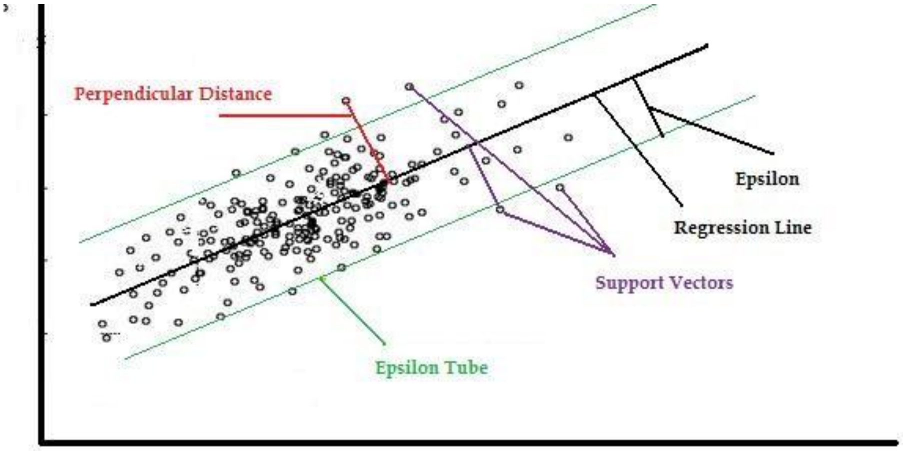
In the figure 3a , we have a regression line with the given value of epsilon.We can find perpendicular distances of each point from the line and find the maximum distance among these.The value of this maximum perpendicular distance is taken as the value of epsilon and do the regression again.

If all the points are within epsilon tube then we say the value of epsilon to Max epsilon and is denoted as ε_0 .

Fig.(3a)



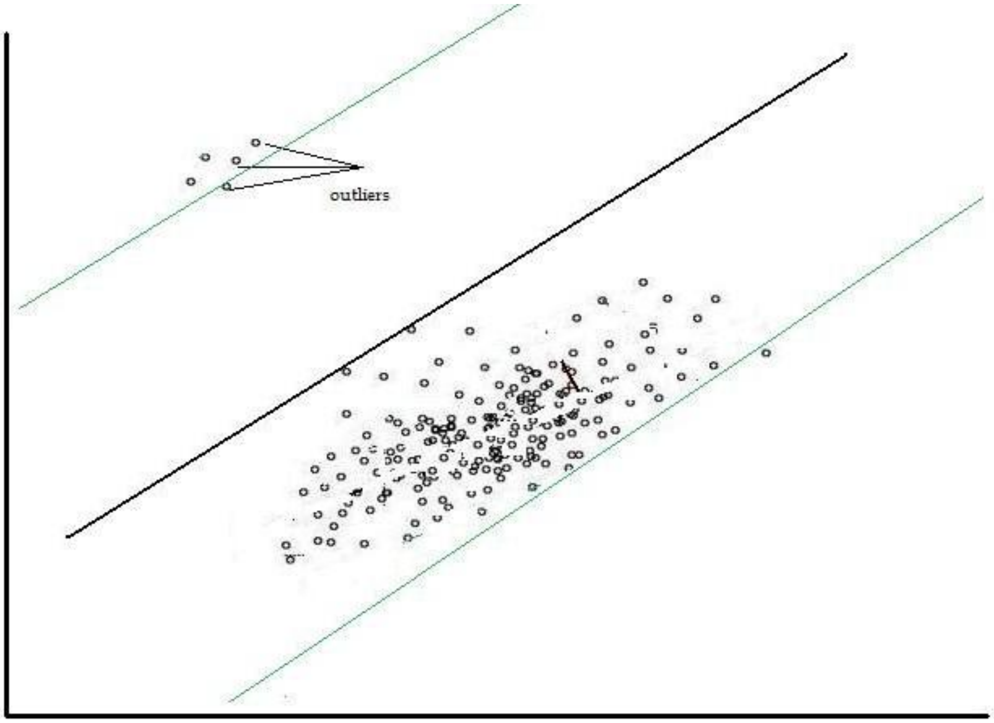
Usual Support Vector Regression When All the points are within epsilon



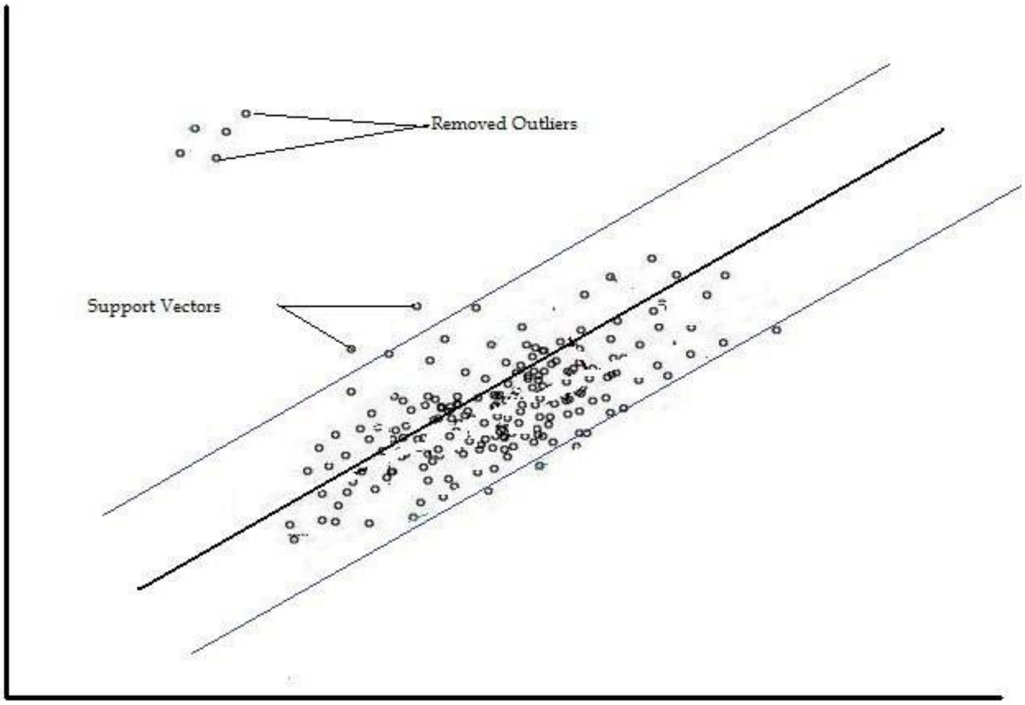
Usual Support Vector Regression withot outlires

Fig (3b)

Suppose the dataset contains outliers then Support Vector Regression will give poor regression.



Usual Support Vector Regression with outliers



SVR After removig outliers

Remarks:

1. In the regression method it is necessary to select both a representative loss function and any additional capacity control that may be required. As we have used ε -insensitive loss function here.

2. We could have added the constraints that $\xi_i \geq 0$ and $\xi_i ' \geq 0$. However, it is not hard to see that the final solution will have that requirement automatically and there is no sense in constraining the optimization to the optimal solution as well. To see this, imagine some ξ_i is negative, then, by setting $\xi_i = 0$ the cost is lower and none of the constraints is violated, so it is preferred.

3. For several reasons (model selection, controlling the number of support vectors, etc.) it may happen that one has to train a SV machine with different regularization parameters *C*, but otherwise rather identical settings. Value of C controls the flatness of the regression line.

Chapter 3: Support Vector Regression for Outlier Removal

3.1 Problem Statement:

An **outlier** is an observation that is numerically distant from the rest of the data. In larger samplings of data, some data points will be further away from the sample mean than what is deemed reasonable. This can be due to incidental systematic error or flaws in the theory that generated an assumed family of probability distribution, or it may be that some observations are far from the center of the data. Outlier points can therefore indicate faulty data, erroneous procedures, or areas where a certain theory might not be valid. However, in large samples, a small number of outliers is to be expected (and not due to any anomalous condition).

Grubbs [17] defined an outlier as: "An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs." Outliers can occur by chance in any distribution, but they are often indicative *either* of measurement error or that the population has a heavy-tailed distribution.

In the case of normally distributed data, roughly 1 in 22 observations will differ by twice the standard deviation or more from the mean, and 1 in 370 will deviate by three times the standard deviation; In a sample of 1000 observations, the presence of up to five observations deviating from the mean by more than three times the standard deviation is within the range of what can be expected, being less than twice the expected number and hence within 1 standard deviation of the expected number.

As we have seen earlier Usual Support Vector Regression method is not good for the data set with outliers and gives a poor regression in result. So how to tackle this type of situation ?

3.2 Suggested Method:

In usual Support Vector Regression technique our goal is to estimate an unknown continuous valued estimation function based on the finite number of noisy samples.

In Support Vector Regression approach, a linear model is constructed in the input space. The linear model f(x,w) is give by

$$f(x,w) = \sum_{i=1}^{n} w_i . x_i + b$$

Where w is the parameter that needs to be determine, b is the bias term.

The quality of estimation is measured by the Loss function. SVR approach uses the ε -insensitive loss function.

$$\text{L(y,f(x,w)} \quad = \quad \begin{cases} 0 & \text{if } |\,\text{y - f(x, w)}\,| < \varepsilon \\ |\,\text{y - f(x, w)}\,| \text{ -}\varepsilon & \text{o.w.} \end{cases}$$

An ε is defined as that if the value of e within the zone, the loss is zero. Otherwise, the loss is the magnitude of the difference between the absolute value of e and ε zone.

The SVR approach performs a linear regression in the input space and tries to reduce model complexity by minimizing $||w||^2$. This can be described by introducing (non –negative) slack variables ξ_i , ξ_i * for i=1,…..n , to measure the deviation of the training samples outside ε -insensitive zone. Thus SVR is formulated as minimization of the following functional:

$$\min_{w,b,\xi_i,\xi_i*} \quad \mathrm{R}\,(w,b\,,\ \xi_i\,,\xi_i\,*)=\mathrm{C}\sum_{i=1}^{n}\ (\xi_i+\xi_i\,*)\ +\ \frac{1}{2}\|\,w\,\|^2$$

Subject to $y_i - \langle w, x_i \rangle - b \le \varepsilon + \xi_i$;

$\langle w, x_i \rangle + b - y_i \le \varepsilon + \xi_i *$;

$\xi_i \geq 0$; $\xi_i * \geq 0$;

The optimization problem can be transformed into the dual problem and its solution is given by

$$f(x) = \sum_{i=1}^{n_{SV}} (\alpha_i - \alpha_i\, *) < x_i, x > +b$$

Where n_{SV} is the number of support vector for the original training data

The points outside the ε -insensitive zone are known as support vectors or outliers. To determine which the support vectors are regarded as outlier, the criteria is build as follows:

If ξ_i or $\xi_i *$ ≥ "4*standard deviation of slack variables", then corresponding support vectors are treated as outliers otherwise Support Vectors. Where the slack variables are nothing but the perpendicular distance of the support vectors from the regression line. i.e. the standard deviation is taken upon the distances of the support vectors.

After getting these outliers, we can remove them from original data set and find the appropriate regression line with the use of SVR.

Main Programming Steps:

```
data1 = textread('x1.txt');%,'','delimiter','\t');
data2 = textread('x2.txt');%,'','delimiter','\t');
data3 = textread('x3.txt');%,'','delimiter','\t');
data4 = textread('x4.txt');%,'','delimiter','\t');
x=[data1 data2 data3 data4];
Y=textread('Y.txt');
y=textread(‘y.txt');
C = 100;
epsilon = .5;
kernel =’linear;
svrreg ( SVR Regression) function calling

function: [xsuport, ysuport ,w, b,newposition,nsupport]=svrreg(x,y,C,epsilon);

           n = length(y);

              %Construct the matrix and a vector

        pp  =  zeros(n,n);
        pp=svmkernel(x,kernel);
        H = pp;
        I = eye(n);
        Idif = [I  -I];
        H = Idif'*H*Idif;
        c = [-epsilon+y ; -epsilon-y];
        A = [ones(1,n)  -ones(1,n) ]';
        b=0;

        [alpha,bias,position]=qp(H,c,A,b,C, x,pp);

        aix=zeros(length(H),1);
        aix(pos)=alpha;
```

```
            alpha=aix;
        w = alpha(newpos)-alpha(n+newpos);
        ysupport = y(newposition);
        nsupport =length(newposition);
        b=bias;
        obj=-0.5*alpha'*H*alpha + c'*alpha;

        function :    [k]=svmkernel(x,kernel,xsup);
        [n1 n2]=size(x);
        [n n3]=size(xsup);
        pp =  zeros(n1,n);
        K=x*xsup';
```

svrval(SVR Value) function calling:

```
        function : f(x)=svmval(x,xsup,w,b,kernel);
        pp=svmkernel(x,kernel,kerneloption,xsup,framematrix,vector,dual);
        y=pp*w+b;
dispf(x); % display function value f(x)
disp(xsup);% display support vector 4 dimensional
disp(ysup);% display corresponding  y value to the support vector
dlmwrite('f(x).txt',f(x),'\n');
x0=ones(400,1);


x=[x0 data1 data2 data3 data4];
yy=textread(‘f(x).txt');
z=x\yy;
% finding the perpendicular distance of the support vector from the line
sq=sqrt(z(1)^2+z(2)^2+z(3)^2+z(4)^2+z(5)^2);
Dist_pts=x*z/sq;
%std(Dist_pts);
c=zeros(2,400);
for i=1:numel(newpos)
   c(1,i)=Dist_pts(newpos(i));
   c(2,i)=newpos(i);
end
std1=std(c(1,:))
j=1;
```

```
   out1=zeros(2,1);
for i=1:numel(c(1,:))
   if c(1,i)>= 4*std1
      out1(1,j)=c(1,i);
      out1(2,j)=c(2,i);
      j=j+1;
   end
end

   output=zeros(2,j);
 output=out1;
% Display the outliers
   disp(output);
```

Chapter 4: Results and Comparisons

4.1 Data set generation with and without outliers

Here we are working on 4 dim artificial data. To generate the data we need x1, x2, x3, x4 and a linear relationship between them. We generate
x1 from a uniform distribution on the unit interval i.e. on the interval [0,1],
x2 from a uniform distribution on the interval [1,4],
x3 from a uniform distribution on the interval [-1,2], and
x4 from a uniform distribution on the unit interval i.e. on the interval [0,1].

Y=2-3*x1+4*x2+x3+0*x4 is the linear relationship between x1,x2,x3 and x4 and our regression function y is given by normal distribution with the mean of Y and variance of 1.

To get some outliers in the dataset we add three points in the data set that are not match with the other points of the dataset.

4.2 Usual Method

Run the multivariate program on the dataset without outliers, then we get the equation of regression hyper plane with coefficients a0=1.8151 a1=-2.9282 a2=4.0346 a3=1.0147 a4=0.2789 and the average distance of the points to the hyper plane is 4.458307.

Run the multivariate program on the dataset with outliers, then we get the regression hyper plane and the average distance of the points to the hyper plane is 4.4034.

4.3 SVR

Run the SVR program on the dataset with outliers with value of epsilon to be 0.5, then we get the equation of regression hyper plane with coefficients a0=2.0307, a1 =-2.9217, a2=3.9882, a3=0.9641, a4=0.1209 and the average distance of the points to the hyper plane is 4.768815 which is quite large in compare to multivariate regression.

4.4 SVR for outlier removal

Now run the Usual SVR program on the dataset with outliers with value of epsilon to be 0.5, then we get regression hyper plane with coefficients a0=2.0307, a1 =-2.9217, a2=3.9882, a3=0.9641, a4=0.1209 the maximum perpendicular distance of the points from the hyper plane which is 7.7887. We find that there are three vectors which are outliers, and then we remove those points and run the SVR on rest of the points. And we find that the equation of the hyper plane has been change and the value of the coefficients are a0=2.0045 , a1=-3.0331 , a2=3.9651, a3=1.0625 , a4=0.2672 and the average distance of the points to the hyper plane is 4.306906 , which is less small in compare to multivariate and usual SVR both.

So therefore if data sets contain outliers and we do usual support vector regression on that dataset then we will get the poor regression hyper plane in compare to multivariate regression. But if we first remove outliers then the line will get shifted to the very fit hyper plane.

Chapter 5. Conclusion and Discussion

In the above approach, we are finding the value of maximum epsilon by using the usual Support Vector Regression Method. And for value of epsilon less than maximum epsilon see the performance. If outliers are not present in the data set, then usual Support Vector do the better regression in compare to Multivariate Regression. But if outliers are present in the dataset then first we will have to remove that outlier by the above method and then do the regression on the inliers data points. So ultimately we are using Support Vector Regression Method twice so its complexity will increase by the factor of 2. SVR performance depends on a good setting of meta-parameters parameters C, ε and the kernel parameters. Parameter C determines the tradeoff between the model complexity (flatness) and the degree to which deviations larger than ε are tolerated in optimization formulation for example, if C is too large (infinity), then the objective is to minimize the empirical risk only, without regard to model complexity part in the optimization formulation.

As mentioned above, this method provides solution gradually. One can develop a different way to combine the twice using of Support Vector Regression method.

References:

[1] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,”Royal Holloway College, London, U.K., Neuro COLT Tech. Rep.TR-1998-030, 1998.

[2] A. J. Smola, B. Schölkopf, and K. R. Müller, “General cost functions for support vector regression,” presented at the ACNN, Australian Congr. Neural Networks, 1998.

[3] A. J. Smola, “Regression estimation with support vector learning machines,”Master’s thesis, Technical Univ. Munchen, Munich, Germany,1998.

[4]] Barnett, V. and Lewis, T.: 1994, Outliers in Statistical Data. John Wiley & Sons., 3rd edition.

[5] Bazaraa M.S., Sherali H.D., and Shetty C.M. 1993. Nonlinear Programming: Theory and Algorithms, 2nd edition, Wiley.

[6] Bunch J.R. and Kaufman L. 1980. A computational method for the indefinite quadratic programming problem. Linear Algebra and Its Applications, pp. 341-370, December.

[7] B. Sch¨olkopf and A. Smola. *Learning with Kernels*. MIT press, 2002.

[8] B. Schölkopf *et al.*, “Comparing support vector machines with Gaussian kernels to radial basis function classifiers,” *IEEE Trans. Signal Processing*,vol. 45, pp. 2758–2765, Nov. 1997.

[9] B. Schölkopf, (2000), “A short tutorial on kernels”, Tutorial given at the *NIPS'00 Kernel Workshop.*

[10] B. Boser, I. Guyon, and V. Vapnik, “A training algorithm for optimal margin classifiers,” presented at the 5th Annu. Workshop Comput. Learning Theory, 1992.

[11] Christopher J. C. Burges, Geometry and invariance in kernel based methods, Advances in kernel methods: support vector learning, MIT Press, Cambridge, MA, 1999

[12] C. Cortes and V. Vapnik, “Support vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[13] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition, ”*Data Mining Knowledge Discovery*, vol. 2, no. 2, 1996.

[14] C.J.C. Burges, (1998), “A tutorial on support vector machines for pattern recognition”, *Knowledge Discovery and Data Mining*,

[15] Debasish Basak, Srimanta Pal and Dipak Chandra Patranabis, “Support Vector Regression”.

[16] Fisher, R.A. (1922). "The goodness of fit of regression formulae, and the distribution of regression coefficients". *J. Royal Statist. Soc.* (Blackwell Publishing) **85** (4): 597–612.

[17] Grubbs, F. E.: 1969, Procedures for detecting outlying observations in samples. Technometrics 11, 1–21.

[18] H. Drucker *et al.*, “Support vector regression machines,” in *Neural information Processing Systems*. Cambridge, MA: MIT Press, 1997, vol. 9.

[19] Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola and Vladimir Vapnik (1997). "Support Vector Regression Machines". *Advances in Neural Information Processing Systems 9, NIPS 1996*, 155-161, MIT Press.

[20] Jinbo Bi and Kristin P. Bennett , “Duality, Geometry, and Support Vector Regression”.

[21] Lindley, D.V. (1987). "Regression and correlation analysis," New Palgrave: A Dictionary of Economics, v. 4, pp. 120–23.

[22] Mangasarian O.L. 1969. Nonlinear Programming. McGraw-Hill, New York.

[23] Marti A. Hearst, Support Vector Machines, IEEE Intelligent Systems, v.13 n.4, p.18-28, July 1998 [doi>10.1109/5254.708428]

[24] McCormick G.P. 1983. Nonlinear Programming: Theory, Algorithms, and Applications. John Wiley and Sons, New York.

[25] Nello Cristianini , John Shawe-Taylor, An introduction to support Vector Machines: and other kernel-based learning methods, Cambridge University Press, New York, NY, 1999

[26] O. Chapelle and V. Vapnik, Model Selection for Support Vector Machines. In Advances in Neural Information Processing Systems*,* Vol 12, (1999)

[27] Renchrer Alvin C., Brigham Young University .Methods of Multivariate Analysis ,Second Edition

[28] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using a support vector machine," in *Proc. NNSP*, 1997, pp.24–26.

[29] Schölkopf B. and Smola A.J. 2002. Learning with Kernels. MIT Press.38

[30] Suykens J.A.K., Vandewalle J., Least squares support vector machine classifiers, Neural Processing Letters, vol. 9, no. 3, Jun. 1999, pp. 293-300.

[31] Vanderbei R.J. 1994. LOQO: An interior point code for quadratic programming.TR SOR-94-15, Statistics and Operations Research, Princeton Univ., NJ.

[32] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[33] V. Vapnik, S. Golowich, and A. J. Smola, "Support vector method for function approximation, regression estimation, and signal processing,"in *Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1997, vol. 9

Appendix

Most algorithms rely on results from the duality theory in convex optimization. For the sake of convenience, briefly review without proof the core results.

Uniqueness Every convex constrained optimization problem has a unique minimum. If the problem is strictly convex then the solution is unique. This means that SVs are not plagued with the problem of *local minima* as Neural Networks.

Lagrange Function The Lagrange function is given by the primal objective function minus the sum of all products between constraints and corresponding Lagrange multipliers. Optimization can be seen as minimization of the Lagrangian wrt the primal variables and simultaneous maximization wrt the Lagrange multipliers, i.e. dual variables. It has a saddle point at the solution. Usually the Lagrange function is only a theoretical device to derive the dual objective function.

Dual Objective Function It is derived by minimizing the Lagrange function with respect to the primal variables and subsequent elimination of the latter. Hence it can be written solely in terms of the dual variables.

Karush–Kuhn–Tucker (KKT) conditions A set of primal and dual variables that is both feasible and satisfies the KKT conditions is the solution (i.e. constraint · dual variable = 0). The sum of the violated KKT terms determines exactly the size of the duality gap (that is, we simply compute the constraint. This allows us to compute the latter quite easily. A simple intuition is that for violated constraints the dual variable could be increased arbitrarily, thus rendering the Lagrange function arbitrarily large. This, however, is in contradition to the saddlepoint property.