UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

UNCERTAINTY AND SENSITIVITY ANALYSIS IN SUPPORT VECTOR MACHINES: ROBUST OPTIMIZATION AND UNCERTAIN PROGRAMMING APPROACHES

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By

JIN PARK
Norman, Oklahoma
2006

UMI Number: 3217533

UMI Microform 3217533

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

UNCERTAINTY AND SENSITIVITY ANALYSIS IN SUPPORT VECTOR MACHINES: ROBUST OPTIMIZATION AND UNCERTAIN PROGRAMMING APPROACHES

A DISSERTATION APPROVED FOR THE
SCHOOL OF INDUSTRIAL ENGINEERING

BY

Dr. Theodore B. Trafalis (Chair)

Dr. P. Simin Pulat

Dr. Mary C. Court

Dr. Suleyman Karabuk

Dr. Michael B. Richman

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Theodore B. Trafalis, for his unwavering support and encouragement throughout my studies at the University of Oklahoma. He has been extremely patient and concerned, even at the most difficult times during my dissertation work. In particular, I would like to express my gratitude to Dr. M. Richman, Dr. P.S. Pulat, Dr. M.C. Court, and Dr. S. Karabuk who have provided careful guidance and advice while serving on my committee throughout my doctoral studies.

None of this work could have been undertaken without the patience and encouragements of my wife, Hyunmin and my children, Inyong and Yong.

I would also like to thank the Republic of Korea Army for giving me the opportunity to study in the United States. Finally, I would like to acknowledge the support of the National Science Foundation under the NSF Grant EIA-0205628.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Noisy or uncertain data are common in machine learning and data mining applications. Noisy data can significantly affect the behavior of data mining and machine learning algorithms. Robust optimization and sensitivity analysis techniques are applied to the support vector machine (SVM) learning problems to develop a noise-immune solution, and suggest new approaches for dealing with noisy data. Perturbations of model parameters are considered as well as perturbation of input data. This approach determines how the levels of noise of data and model parameters influence the SVM solution, both in linear and nonlinear problems. Probability and scenario constrained approaches are also examined as alternatives to the robust optimization approach. Several examples illustrate the proposed methods. An application to real time traffic data for the prediction of the speed of a vehicle is also discussed. Tornado data analysis is illustrated in a probability constrained approach as well.

CHAPTER 1. Introduction

1.1 Overview and Research Objectives

The objective of this dissertation is to investigate robust optimization and sensitivity analysis techniques applied to the support vector machine (SVM) learning problem and suggest new approaches for dealing with noisy data. An additional objective is to develop a scenario constrained programming approach as an alternative to the robust optimization approach. Sensitivity analysis in machine learning can be used to show how the machine learning model performs when the model is changed. Specifically, its aim is to determine how much the variation of the input can influence the output of the learning machine. Sensitivity analysis is an issue for machine learning because imperfect datasets occur frequently in practice. Several researchers applied sensitivity analysis to the optimization problem using the perturbation of input data (Bonnans and Shapiro, 2000). However, sensitivity analysis has not been extensively studied in machine learning.

Similarly, a lot of researchers have dealt with data uncertainties, and many

different robust optimization approaches have been suggested. Ben–Tal and Nemirovski (1998) introduced bounded uncertainty convex sets to describe uncertain coefficients in mathematical programming. By using bounded convex uncertainty sets, such as ellipsoidal uncertainty sets, they developed a robust optimization approach for linear programming (LP), semi-definite programming (SDP) and other problems. Based on minimax optimization arguments, they developed a robust counterpart (RC) approach for convex programming. Chinneck and Ramadan (2000) considered LP problems with interval coefficients.

Calafiore and Campi (2005) introduced an uncertain convex program (UCP) using the concept of an ε-level solution. The ε-level represents the risk of the constraint violation. Since the two main approaches for the uncertainty constrained optimization problem, robust optimization and probability constrained optimization, lead to a computationally intractable problem formulation, they considered a randomized scenario approach. This approach is based on constraint sampling with a finite set of *N* constraints, which needs a sufficient number of constraints to represent the whole set of constraints. They have addressed the problem of how many samples (scenarios) need to be drawn in order to guarantee that the resulting randomized

solution violates only a “small portion” of the constraints.

Bertsimas et al. (2004) described an approach using a general norm to define the uncertainty set and derived probabilistic guarantees on the feasibility of a robust optimal solution with respect to a general and dual norm, respectively.

Recently a lot of attention has been given to SVM (Vapnik, 1995). The SVM approach consists of finding the hyperplane that separates two sets of points in such a way that the distance between the hyperplane and the nearest point of each of the data set is maximum. The resulting SVM learning convex optimization problem provides the “best” feasible solution in terms of generalization behavior for the separation constraints with respect to w and b, where w is the vector defining the separation hyperplane, and b is the offset of this hyperplane. The SVM approach has been developed for input data without noise. An interesting problem is to investigate the behavior of the SVM solution with noisy (perturbed) data and model parameters.

There are two ways that randomness can be applied to machine learning algorithms; the first one originates from the sampling procedure to construct the learning set, and the second one comes due to noise in the observations and parameters. Bousquet and Elisseff (2002) focused on sampling randomness and how changes in the

learning set can influence the function produced (discriminant or regressor). Trafalis and Alwazzi (2003a, 2003b), and Trafalis and Gilbert (2006) investigated a robust optimization approach with bounded perturbations of the input training data applied to support vector machines (SVMs). Ghaoui et al. (2003) considered binary, linear classification problems where the data points are unknown but bounded within given hyper-rectangles. They designed a robust classifier by minimizing the worst-case value of given loss functions such as hinge loss, negative log likelihood function, and minimax probability machines (MPM) loss function.

In the optimization literature, generally only perturbations of input data are considered. Investigation of the stability of SVM solutions with respect to changes of input data and model parameters is of concern in practical applications. We build on previous research by Trafalis and Alwazzi (2003a, 2003b) by considering perturbations both of input data and model parameters. The motivation for our analysis is to design robust machine learning algorithms that are “immune” to noise of inputs and parameters. We also develop a scenario constrained optimization approach as an alternative to robust optimization approaches.

1.2 Organization of the Dissertation

Chapter 2 deals with the basic concepts of robust optimization and some optimization methods which are related to this dissertation. The basic concepts of Support Vector Machines (SVMs) and our novel approaches are outlined in chapter 3, and a new approach with perturbations of input data and perturbations of the SVM model parameters are also explored. Computational results for a classification and regression problem are discussed in chapter 4. Probability constrained programming and a scenario-based approach are discussed in chapter 5. Examples and computational results are also shown in the same chapter. Lastly, chapter 6 concludes the dissertation and describes future work.

CHAPTER 2. Literature Survey

Several formulations related to robust optimization and sensitivity analysis are outlined in this chapter. In most real applications, an optimal solution is affected by the structure of the data set. Very often the data may be inaccurate or missing.

2.1 Robust Optimization of Linear Programming

Ben–Tal and Nemirovski (1999, 2000) investigated LP problems with uncertain data. The following linear program:

$$\begin{aligned} &\min \ c^T x \\ &s.t \quad Ax \le b \end{aligned} \tag{2.1}$$

is assumed to be uncertain in the sense that the data set A and b are defined through uncertainty convex bounded sets in the space of $m \times n$ matrices and R^n, respectively. The key idea of their study about convex optimization problems with uncertainty is that the data are not accurately specified. They defined the uncertainty set U in the space of data, where the uncertain parameters belong to a bounded and convex set. Therefore, the resulting optimization problem becomes:

$$\begin{aligned} &\min \ c^T x \\ &s.t \quad Ax \le b \quad \forall (A,b) \in U \end{aligned} \tag{2.2}$$

The above problem is a semi-infinite optimization problem (Calafiore and Campi, 2005). Problem (2.2) is a robust counterpart (RC) of the linear programming problem (2.1). They proposed two ways to implement this method: “unknown-but-bounded” uncertainty and “random symmetric” uncertainty. They developed robust convex programs corresponding to some of the most important generic convex problems. The robust formulation of problems such as linear programming, semi-definite programming and others, are either exactly or approximately tractable problems, which can be solved by efficient polynomial time algorithms. An example of an efficient algorithm that can be applied is a polynomial time interior point method when the set *U* is an ellipsoidal uncertainty set.

In the case of ellipsoidal uncertainties, problem (2.2) will be as follows:

$$\begin{aligned} &\min_{x \in R^n} \ c^T x \\ &s.t \quad \mathrm{A}(\delta)\,\mathrm{x} \le \mathrm{b}, \ \ \delta \in \Delta \end{aligned} \tag{2.3}$$

where we assume that $\mathrm{A}(\delta)$ is affine in δ and the set Δ is the direct product of ellipsoids (Calafiore and Campi, 2005). We also assume each constraint row belongs to an ellipsoid. That is,

$$a_i(\delta) = \hat{a}_i + E_i\delta_i, \quad \|\delta_i\|_2 \le 1, \quad i = 1,\Lambda\ ,m \tag{2.4}$$

where $\hat{a}_i \in R^n$ is the center of the ellipsoid and $E_i = E_i^T \in R^{n\times n}$ is the shape matrix of the ellipsoid describing the variation in a_i. Then, the constraints of (2.3) become $\hat{a}_i^T x + \delta_i^T E_i \mathrm{x} \le \mathrm{b}_i$. From the ellipsoidal uncertainty description, the constraints hold for all $\delta \in \Delta$ if and only if:

$$\max_{\|\delta_i\|_2 \le 1} \hat{a}_i^T x + \delta_i^T E_i \mathrm{x} \le \mathrm{b}_i . \tag{2.5}$$

Therefore, the robust formulation of the LP problem (2.3) can be expressed as the second order cone (SOC) program (Lobo et al. (1998)) when Euclidean norm is considered as follows:

$$\begin{aligned} &\min_{x\in R^n} c^T x \\ &s.\mathrm{t} \quad \hat{a}_i^T x + \|E_i x\|_2 \le b_i,\ i = 1,\Lambda\ ,m \end{aligned} \tag{2.6}$$

Second order cone programming (SOCP) is a convex nonlinear optimization problem that includes linear and quadratic programs (Lobo et al., 1998, Alizadeh and Goldfarb, 2003). Lobo et al. (1998) showed how SOCP can be applied to solve the robust convex optimization problems that the uncertainty in the data set is explicitly represented.

The LP (2.1) can be also considered in a statistical framework. If we assume

that the parameter a_i follows a normal distribution, with mean and covariance $(\hat{a}_i, \Sigma_i)$, then the constraints can be expressed as:

$$\Pr[a_i^T x \le b_i] \ge \beta, \tag{2.7}$$

where β represents a (probability) confidence level. This analysis is discussed in Ben-Tal and Nemirovski (2000), Oustry et al. (1998), and Lobo et al. (1998). This robust LP can be expressed as the following SOCP problem:

$$\begin{aligned} &\min \ c^T x \\ &s.t \ \ \hat{a}_i^T x + \Phi^{-1}(\beta)\left\|\Sigma_i^{1/2} x\right\|_2 \le b_i, \ i = 1, \Lambda \ , m \end{aligned} \tag{2.8}$$

where, β is a given confidence level, and Φ is the cumulative distribution function (CDF).

2.2 Robust Classification with Interval Data

Ghaoui et al. (2003) considered uncertain data defined within a specific uncertainty interval defined as:

$$\mathrm{X}(\rho) = \left\{ Z \in R^{n \times N} : \mathrm{X} - \rho\Sigma \le Z \le \mathrm{X} + \rho\Sigma \right\}, \tag{2.9}$$

where X, Σ and ρ describe an interval matrix model for a $n \times N$ data matrix (n

dimensions, N nominal data points). They considered N hyper-rectangles of dimension n in the input space R^n. The data of this model is a set of points $\{(x_i, y_i)\}$ (training data), where $x_i \in R^n$ and $y_i = \pm 1$. The objective of this approach is to find a classification hyperplane $w^T x + b = 0$, where $w \in R^n - \{0\}$, b is a scalar, and x is a testing point that needs to be classified. The method of robust classification with interval data minimizes a worst-case loss function. For the hinge loss function, (Cristianini and Shawe-Taylor, 2000), they used the worst-case loss function. This is defined as:

$$\begin{aligned} L_{SVM}(w,b) &= \max_{Z \in \mathrm{X}(\rho)} \sum_{i=1}^{N} (1 - y_i (w^T z_i + b))_+ \\ &= \sum_{i=1}^{N} (1 - y_i (w^T x_i + b) + \rho {\sigma_i}^T |w|)_+ \end{aligned} \tag{2.10}$$

where $(\cdot)_+$ represents the positive part of a scalar $(\cdot)$.

To illustrate the above approach, an example is discussed in chapter 4. It is a variation of the AND example for the robust LP problem with interval data. We also use the same example to illustrate the primal SVM problem. (Vapnik, 1998).

2.3 Uncertain Convex Programs

Calafiore and Campi (2005) considered the following uncertain convex

program:

$$UCP : \{ \min_{x \in X \subseteq R^n} c^T x \; ; \; f(x, \delta) \leq 0, \, \delta \in \Delta \} \tag{2.11}$$

where $x \in X$ is the optimization variable, X is convex and closed, and the function f is convex in x for all $\delta \in \Delta$, where Δ is a parameter set.

A robust formulation (RCP) for UCP is as follows:

$$\begin{aligned} RCP : & \min_{X \in R^n} c^T x \\ s.t \quad & x \in X \text{ I } \Omega \end{aligned} \tag{2.12}$$

$$\text{where, } \Omega = \prod_{\delta \in \Delta} \{x : f(x, \delta) \leq 0\} \quad \text{and} \quad X \text{ I } \Omega \neq \phi$$

Let x be a candidate solution for UCP. The violation probability is defined as:

$$V(x) = P\{\delta \in \Delta : f(x, \delta) > 0\} \tag{2.13}$$

Let $\varepsilon \in [0,1]$. We say that $x \in X$ is an ε-level robust feasible solution if $V(x) \leq \varepsilon$.

In order to solve RCP, they proposed to collect N randomly chosen samples and solved the following optimization problem:

$$\begin{aligned} SCP_N : & \min_{x \in R^n} c^T x \\ s.t \quad & f(x, \delta^{(i)}) \leq 0, \; i = 1, \Lambda \; , N \\ & x \in X \end{aligned} \tag{2.14}$$

They addressed the question of how many samples (scenarios) need to be drawn in order to guarantee that the resulting randomized solution violates only a “small

portion" of the constraints. They showed that the required number of constraints N is as follows:

$$N \geq \frac{n}{\varepsilon\beta} - 1 \tag{2.15}$$

Specifically, they proved:

Proposition (Calafiore and Campi, 2005))

Fix two real numbers $\varepsilon \in [0,1]$ (level parameter) and $\beta \in [0,1]$ (confidence parameter) and let

$$N \geq \frac{n}{\varepsilon\beta} - 1$$

Then, with probability no smaller than $1-\beta$, the randomized problem SCP_N returns an optimal solution $\hat{x}_N$ which is ε-level robust feasible. They also showed the following theorem:

Theorem 1 (Calafiore and Campi, 2005)

Let $\hat{x}_N$ be the (unique) solution to SCP_N. Then

$$E_{P^N}[V(\hat{x}_N)] \leq \frac{n}{N+1} \tag{2.16}$$

where n is the size of x, and p^N is the probability measure in the space Δ^N of the multi-sample extraction $\delta(1), \Lambda, \delta(N)$. Therefore, the average probability of

violation of $\hat{x}_N$ is proportional to the dimensionality of the optimization variable x and goes to zero linearly with the number N of sampled constraints.

In order to illustrate the above approach for problem (2.3), we consider 4 data points {(-1, 0), (0, -1), (1, 0), (0, 1)} that define the rows of matrix *A*, where $b = [0 \;\; 0\; 1 \;\; 1]^T$, and $c = [-1 \;\; -1]$. The problem is given by the linear program:

$$\begin{aligned} &\min_{x \in R^n} \;\; c^T x \\ &s.t \quad A(\delta^{(i)})\, x \le b, \quad i = 1, \Lambda \;, N \end{aligned} \tag{2.17}$$

Table 2. 1 Robust LP solution with different probability distributions of the data set.

	Uni(0,1)	N (0,1)	Exp(1)	Exp(0.3)	Exp(0.5)	N(0.5, 0.1)	N(-0.5, 0.1)
X1	0.8830	1.0139	0.8917	0.9157	0.9015	0.8462	1.2236
X2	0.8823	1.0141	0.8904	0.9153	0.9001	0.8462	1.2242

We choose probabilistic levels $\varepsilon = 0.01, \;\; \beta = 0.01$. The number *N* of randomized constraints must be $N \ge \frac{n}{\varepsilon\beta} - 1 = \frac{2}{0.01 * 0.01} - 1 = 19{,}999$. The solutions (Table 2.1) depend on the probability distribution of the data set.

2.4 Robust Optimization with Constraints

Robust optimization and probability constrained optimization are the main approaches related to the handling of the uncertainties. The probability constrained optimization problems consider probability distributions on the constraints with specific confidence levels. However, the probability constrained optimization problem, in general, is not easily solved, and the constraints of the problem cannot guarantee convexity in general.

Ben-Tal and Nemirovski (1996) and Oustry et al. (1998) suggested an SOCP problem formulation for the robust LP problem. Bhattacharyya (2004) and Bhattacharyya et al. (2004) developed a robust formulation with a normal distribution noisy model.

CHAPTER 3. Methodology

3.1 Support Vector Machines (SVMs)

Support vector machine (SVM) is a statistical learning system based on the concept of the maximum margin separation. SVM can handle nonlinear separation problems by mapping the input space into a high dimensional feature space where it constructs an optimal maximum margin hyperplane (Vapnik, 1995). More specifically, we can transform a nonlinear separation problem in the input space into a linear problem in the feature space by use of the kernel function. Then the nonlinear problem can be solved linearly in the feature space. SVM can handle complex nonlinear problems such as pattern recognition, regression, and feature extraction, with excellent generalization properties.

The choice of an appropriate kernel function is a main issue for the SVM algorithms. Linear function, polynomial function, radial basis function (RBF), and sigmoid (tangent hyperbolic) function are well-known kernel functions frequently used by researchers, that satisfy Mercer's Theorem (Vapnik, 1998 and Haykin, 1999). The

following kernel functions are frequently used in the SVM literature:

- Linear function: $K(x_i, x_j) = x_i^T x_j$,
- Polynomial function: $K(x_i, x_j) = (\gamma\, x_i^T x_j + p)^d$,
- Radial Basis function: $K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2)$,
- Sigmoid function: $K(x_i, x_j) = \tanh(\gamma\, x_i^T x_j + p)$

where, *d* is degree, *p* is offset, and γ is a scale parameter.

If the kernel function is selected properly, the SVM can provide a solution with good generalization properties. Note that it is not necessary to compute the feature map. This is expressed implicitly through the kernel function. Therefore, the kernel method easily transforms nonlinear problems into linear problems. As a result, linearly non-separable data in the input space become linearly separable in the feature space.

3.1.1 Linearly Separable Case

The use of the SVM model is considered in the case of data points that can be linearly separated. A set *S* of points $x_i \in R^n$ is assumed, where each x_i belongs to either one of two different classes defined by a label $y_i \in \{-1, 1\}$. The objective is to

find an optimal hyperplane that divides a set S leaving all the points of the same class on the same side while maximizing the minimum distance between the two classes and the hyperplane (Cristianini and Shawe-Taylor, 2000 and Vapnik, 1995). Figure 3.1 shows a linearly separable optimal hyperplane between two classes.

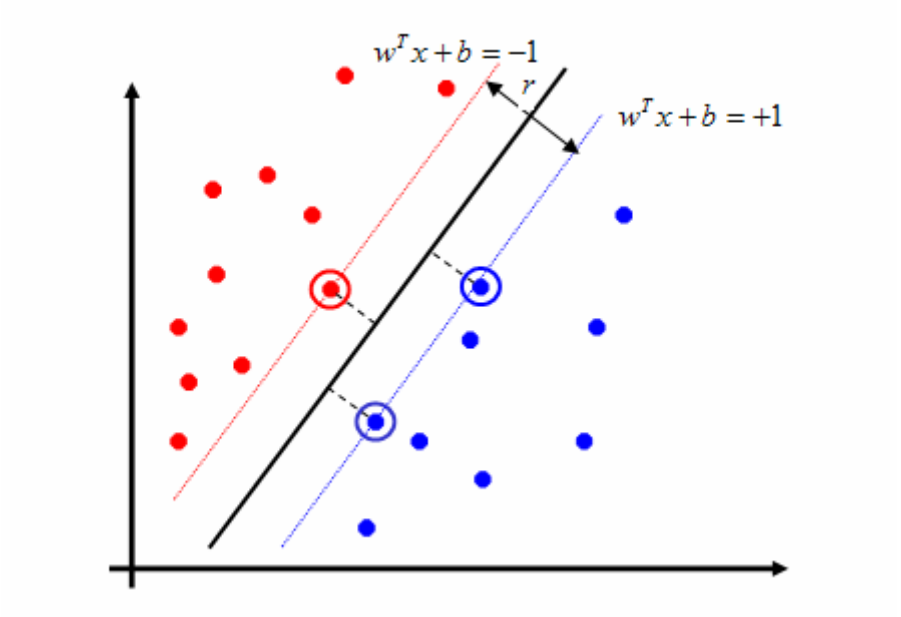

Figure 3. 1 Linearly separable optimal hyperplane. Maximize distance between two parallel supporting planes ($w^T x + b = \pm 1$). The distance (margin) between the two classes is $r = \frac{2}{\|\mathbf{w}\|}$.

Definition 1. The set S is linearly separable if there exist a $w \in R^n$ and $b \in R$ such that

$$\begin{aligned} w^T x_i + b &\geq 1 \qquad \text{if } y_i = +1 \\ w^T x_i + b &\leq -1 \qquad \text{if } y_i = -1 \end{aligned} \tag{3.1}$$

The SVM, or maximal margin classifier formulation, with l data points in the n-dimensional space can be written in primal form as follows:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 \tag{3.2}$$

$$s.t \quad y_i\,(w^T x_i + b) \geq 1, \qquad i = 1,\Lambda\ ,l \tag{3.3}$$

Since $\|w\|^2$ is convex, w can be attained by the use of Lagrangian function:

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i [y_i \cdot (w^T x_i + b) - 1]. \tag{3.4}$$

To find the saddle point of the Lagrangian function, one has to minimize equation (3.4) over w and b and maximize it with respect to the Lagrange multipliers $\alpha_i > 0$. The optimality condition must satisfy the following conditions:

$$\frac{\partial}{\partial b} L(w,b,\alpha) = \sum_{i=1}^{l} a_i y_i = 0 \tag{3.5}$$

$$\frac{\partial}{\partial w} L(w,b,\alpha) = w - \sum_{i=1}^{l} \alpha_i y_i x_i = 0 \tag{3.6}$$

By substituting equations (3.5) and (3.6) into (3.4), the primal optimization problem becomes:

$$\max \ W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j < x_i \cdot x_j > \tag{3.7}$$

with $\alpha_i \geq 0,\ i = 1,...,l.$ Under constraint (3.5) this problem can be solved by using quadratic programming (Bazaraa and Shetty, 1979). The optimal separating

hyperplane will be determined by *w* and *b*. From equation (3.6) and Kuhn-Tucker conditions, we have:

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{3.8}$$

$$\alpha_i \cdot [y_i (w \cdot x_i + b) - 1] = 0 \tag{3.9}$$

Note that the Lagrange multipliers α_i are always non-negative. The data points that correspond to $\alpha_i > 0$ play an important role in the determination of the optimal separating hyperplane. The weight vector *w* is determined by those points, which are called "*support vectors*". The decision function of the primal problem can be written as:

$$f(x) = sign(\sum_{i=1}^{l} y_i \alpha_i < x_i \cdot x > + b). \tag{3.10}$$

3.1.2 Linearly Non-separable Case (Soft Margin Optimal Hyperplane)

If the data are not linearly separable, then constraints (3.3) might not be feasible. In order to compensate for the misclassification error, Vapnik (1995) introduced non-negative slack variables t_i to address the problem of infeasibilities

and the cost regularization parameter C as a weight for misclassification errors. Therefore, in addition to maximizing the margin we need to minimize the sum of misclassification errors, that is $\sum_{i=1}^{l} t_i$.

The constraints of (3.3) can be modified as in equation (3.12), and the resulting optimization problem is as follows:

$$\min \quad \frac{1}{2}\| w\|^2 + C\sum_{i=1}^{l} t_i \tag{3.11}$$

$$s.t \quad y_i(w^T x_i + b) \geq 1 - t_i, \quad t_i \geq 0, \quad i = 1, \Lambda\ , l \tag{3.12}$$

Figure 3.2 illustrates the linearly non-separable (soft margin) optimal hyperplane, which

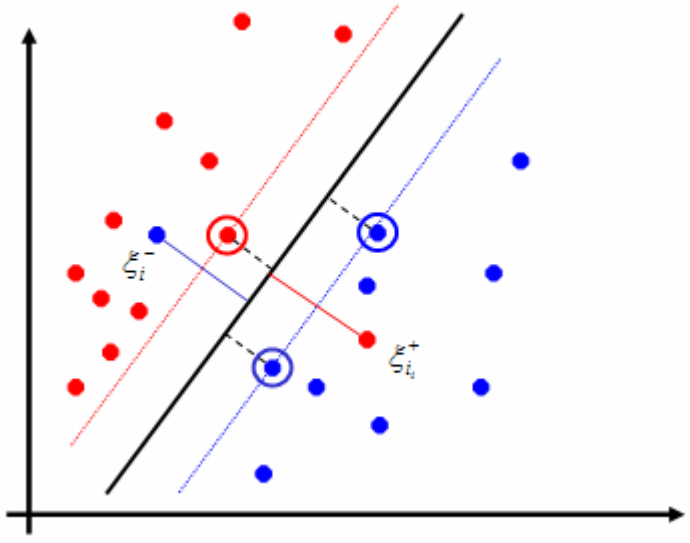

Figure 3. 2 Linearly non-separable optimal hyperplane

has misclassified data points. The parameter C in equation (3.11) is provided by the user and controls the trade off between minimizing the training set error and maximizing the margin.

As shown in (3.4), the above problem can be transformed into the unconstrained problem by introducing the following Lagrangian function (Vapnik, 1998),

$$\Phi(w,b,\alpha,\beta,\xi) = \frac{1}{2}\| w \|^2 + C(\sum_{i=1}^{l}\xi_i) - \sum_{i=1}^{l}\alpha_i[y_i(w\cdot x_i + b) - 1 + t_i] - \sum_{i=1}^{l}\beta_i t_i \quad (3.13)$$

where α, β are the Lagrange multipliers. The optimality condition must satisfy the following equations:

$$\frac{\partial\Phi}{\partial w} = w - \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i = 0 \quad (3.14)$$

$$\frac{\partial\Phi}{\partial b} = \sum_{i=1}^{l}\alpha_i y_i = 0 \quad (3.15)$$

$$\frac{\partial\Phi}{\partial \xi} = \alpha_i + \beta_i = C \quad (3.16)$$

From the above conditions, we derive:

$$w = \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i \quad (3.17)$$

$$\sum_{i=1}^{l}\alpha_i y_i = 0 \quad (3.18)$$

Substituting (3.17) and (3.18) into (3.13), the dual form of the problems becomes:

$$\begin{aligned} \max\ & W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ s.t\ & \quad 0 \le \alpha_i \le C. \end{aligned} \tag{3.19}$$

The upper bound of α_i is *C*, which is the only difference from the linearly separable case. The soft margin parameter *C* also affects the slack variable t_i. When α_i is less than *C*, the slack variable is t_i must be zero by Karush-Kuhn-Tucker (KKT) complementary slackness condition.

3.1.3 Kernel Functions for Nonlinear Support Vector Machines

The idea of a kernel method is based on mapping a data set from input space into feature space. All we need is the inner product in the feature space. Suppose we map the data into some higher dimensional feature space *H*, through the mapping φ. By replacing the inner product with a kernel function K, we can perform a non-linear mapping into a high dimensional feature space. Then the optimization problem of Equation (3.7) can be transformed as follows:

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j), \tag{3.20}$$

where the kernel $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) = < x_i \cdot x_j >$.

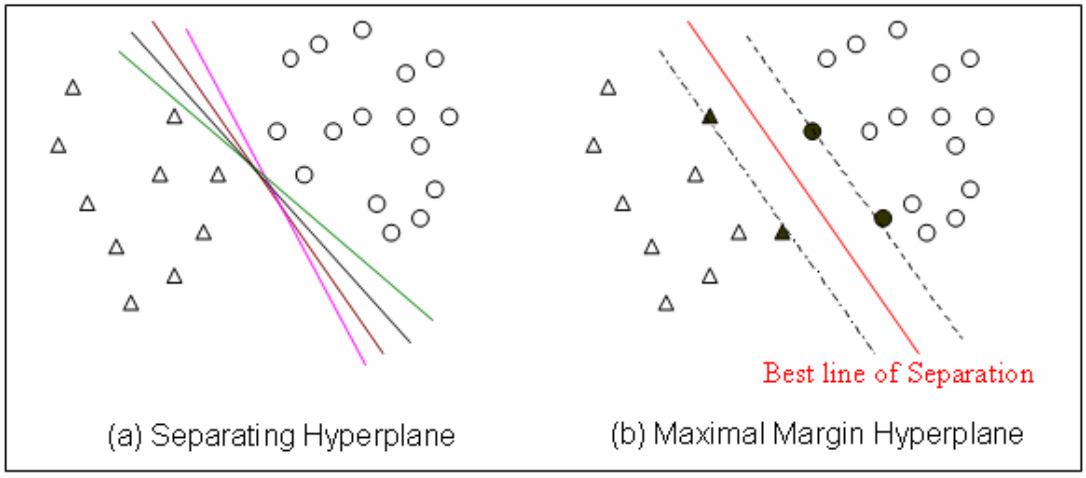

Figure 3. 3 Illustration of separating hyperplane and maximal margin hyperplane.

Figure 3.3 (b) illustrates the SVM solution as the best line of separation. The best margin is called maximal margin (Cortes et. al. 1995). From equation (3.13), (3.17), and (3.18), we can transform the primal optimization formulation (3.11) into the generalized kernel formulation as follows:

$$\begin{aligned} &\min_{\alpha,b,t} \ \frac{1}{2}\alpha^T \tilde{K}\alpha + C\sum_{i=1}^{l} t_i \\ &s.t \\ &\quad \tilde{K}_i\alpha + y_i b \geq 1 - t_i \\ &\quad \alpha_i \geq 0, \quad i = 1,\Lambda\ ,l \end{aligned} \tag{3.21}$$

where $\tilde{K}_{ij} = y_i y_j K(x_i, x_j)$ and K_i is the i-th row of $\tilde{K}$.

Next we consider sensitivity analysis and robust optimization techniques applied to the SVMs' maximal margin classifier.

3.2 Sensitivity Analysis and Robust Optimization

Sensitivity analysis and robust optimization methodology are slightly different concepts with respect to data perturbations. Sensitivity analysis is focused on how much the optimal solution to a perturbed problem can differ from the one of the nominal problems. However, using the robust optimization methodology we are interested in finding a feasible solution to the nominal problem that satisfies the constraints of the perturbed problem for every realization of a bounded perturbation (Ben-Tal and Nemirovski, 2000). A lot of research has dealt with data perturbations for the optimization problem. In this research we also deal with perturbations of parameters of the SVM model extending previous research by Trafalis and Alwazzi (2003), and Trafalis and Gilbert (2006). Now we consider three cases of perturbation for data and parameters respectively. Specifically perturbations of input data, perturbation of parameters and perturbation of both data and parameters, are considered.

3.2.1 SVM for Classification

In this chapter we build up our models. We begin with the perturbation of input data for the SVM classification problem, and then investigate how to handle perturbation of parameters and finally we develop a SVM model where both parameters and data are perturbed. Extension of our models for the regression problem is also discussed.

Case 1: Perturbation of Data

The first case of perturbation of data is to make a slight change Δx_i in the input data ($x \to x + \Delta x_i$). In the real world, the noise of measurements always exists due to several reasons such as experimental errors, and missing values. It is assumed that the perturbations are bounded according to our prior information. The constraint of the primal formulation (3.2) becomes:

$$y_i \ [(w^T (x_i + \Delta x_i) + b] \geq 1 - t_i$$

$$\Rightarrow \ y_i \ [(w^T x_i + w^T \Delta x_i + b] \geq 1 - t_i \qquad (3.22)$$

We know that constraint (3.22) holds for Δx_i bounded ($\|\Delta x_i\| \leq \eta$) in a robust way if and only if the minimum of the left hand side satisfies the following:

$$\min_{\|\Delta x_i\| \le \eta} y_i \quad [(w^T x_i + w^T \Delta x_i + b] \ge 1 - t_i \tag{3.23}$$

Then (3.23) becomes:

$$y_i \quad (w^T x_i + b) + \min_{\|\Delta x_i\| \le \eta} y_i w^T \Delta x_i \ge 1 - t_i \tag{3.24}$$

By Cauchy-Schwarz inequality, we have:

$$\left| y_i w^T \Delta x_i \right| \le \left| y_i \right| \cdot \|w\| \cdot \left\| \Delta x_i \right\| \le 1 \cdot \eta \cdot \|w\| .$$

Therefore, $y_i w^T \Delta x_i$ is bounded as follows:

$$-\eta \|w\| \le y_i w^T \Delta x_i \le \eta \|w\| \qquad . \tag{3.25}$$

Thus the minimum of $y_i w^T \Delta x_i$ is $-\eta \|w\|$. Now the primal problem becomes:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} t_i \\ s.t \quad & y_i \quad (w^T x_i + b) - \eta \|w\| \ge 1 - t_i \\ & i = 1, \mathrm{L} \ , l \end{aligned} \tag{3.26}$$

This problem coincides with the original problem (3.2) when $\eta = 0$ and is a Second Order Cone Programming (SOCP) problem (Lobo et al., 1998) which can be solved by a primal-dual interior point method (Wright, 1997).

Case 2: Perturbation of Parameters

In the case 2 of perturbation of parameters, we consider a change of the

weight vector Δw for problem (3.2) ($w \to w + \Delta w$). The primal constraint of (3.22) becomes:

$$y_i \quad [(w + \Delta w)^T x_i + b] \geq 1 - t_i$$

$$\Rightarrow \quad y_i \quad [w^T x_i + \Delta w^T x_i + b] \geq 1 - t_i \tag{3.27}$$

Note that w is robust feasible with respect to bounded perturbations of the vector w ($\|\Delta w\| \leq \eta$) if and only if for every $i = 1, \Lambda\ , l$

$$\min_{\|\Delta w\| \leq \eta} y_i \quad [w^T x_i + \Delta w^T x_i + b] \geq 1 - t_i . \tag{3.28}$$

Then (3.28) becomes,

$$y_i \quad (w^T x_i + b) + \min_{\|\Delta w\| \leq \eta} y_i \Delta w^T x_i \geq 1 - t_i \tag{3.29}$$

By Cauchy-Schwarz inequality, again we have:

$$\left| y_i \Delta w^T x_i \right| \leq \left| y_i \right| \cdot \|\Delta w\| \cdot \|x_i\| \leq 1 \cdot \eta \cdot R.$$

Note that we assume that $\|x_i\| \leq R$. Then $y_i \Delta w^T x_i$ is bounded as follows:

$$-\eta R \leq y_i \Delta w^T x_i \leq \eta R \tag{3.30}$$

Therefore, the minimum of $y_i \Delta w^T x_i$ is equivalent to $-\eta R$.

Hence the primal problem becomes,

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} t_i \\ s.t \quad & y_i \quad (w^T x_i + b) - \eta R \geq 1 - t_i \\ & i = 1, \mathrm{L} \ , l \end{aligned} \tag{3.31}$$

Case 3: Perturbation of Input Data and Parameters

In this case, we consider perturbation of data and parameters simultaneously ($x \to x + \Delta x_i$, $w \to w + \Delta w$). The constraint of (3.2) becomes,

$$y_i \quad [((w + \Delta w)^T (x_i + \Delta x_i) + b] \geq 1 - t_i$$

$$\Rightarrow \quad y_i \quad [(w^T x_i + w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i) + b] \geq 1 - t_i$$

$$\Rightarrow \quad y_i \quad [(w^T x_i + b) + w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)] \geq 1 - t_i . \tag{3.32}$$

We know that the constraint of (3.32) holds for Δx and Δw if and only if:

$$\min_{\|\Delta x_i\| \leq \eta_1, \|\Delta w\| \leq \eta_2} y_i \quad [(w^T x_i + b) + w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)] \geq 1 - t_i$$

$$\Rightarrow \quad y_i \quad (w^T x_i + b) + \min_{\|\Delta x_i\| \leq \eta_1, \|\Delta w\| \leq \eta_2} y_i [w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)] \geq 1 - t_i$$

By Cauchy-Schwarz inequality, we have:

$$\begin{aligned} -\eta_1 \|w\| \leq y_i w^T \Delta x_i \leq \eta_1 \|w\| \\ -\eta_2 R \leq y_i \Delta w^T x_i \leq \eta_2 R \end{aligned}, \tag{3.33}$$

where we assume $\|x_i\| \leq R$. Similarly by Cauchy-Schwarz, we have:

$$\left| y_i (\Delta w)^T (\Delta x_i) \right| \leq |y_i| \cdot \|\Delta w\| \cdot \|\Delta x_i\| \leq 1 \cdot \eta_1 \cdot \eta_2 .$$

Hence,

$$\begin{aligned}\left|y_i[w^T\Delta x_i+\Delta w^T x_i+(\Delta w)^T\Delta x_i]\right| &\le \left\|w^T\Delta x_i\right\|+\left\|\Delta w^T x_i\right\|+\left\|(\Delta w)^T\Delta x_i\right\| \\ &\le \eta_1\|w\|+\eta_2 R+\eta_1\eta_2\end{aligned} \tag{3.34}$$

Therefore,

$$\min_{\|\Delta x_i\|\le\eta_1,\|\Delta w\|\le\eta_2} y_i[w^T\Delta x_i+\Delta w^T x_i+(\Delta w)^T\Delta x_i] \;=\; -\eta_1\|w\|-\eta_2 R-\eta_1\eta_2 \tag{3.35}$$

Finally, by substituting the above result in problem (3.29), the maximal margin classifier problem considering uncertainty will be as follows:

$$\begin{aligned}\min_{w,b}\quad & \frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}t_i \\ s.t\quad & y_i\ (w^T x_i+b)-\eta_1\|w\|-\eta_2 R-\eta_1\eta_2\ge 1-t_i \\ & i=1,\mathrm{L}\ ,l\end{aligned} \tag{3.36}$$

3.2.2 SVM for Regression

Since Vapnik [27] introduced the ε-insensitive loss function, the support vector regression (SVR) problem has been generalized for function approximation and forecast (Schölkopf and Smola, 2002). Let's consider a set of training data $\{(x_1,y_1),....,(x_l,y_l)\}$, where each $x_i\in R^n$ represents a point in the input space of the sample set and has a corresponding target value $y_i\in R$. In the regression problem,

the y_i are continuous real-valued outputs. The objective of the regression problem is to find a function from the training data that predicts future values. The support vector regression formulation with an ε-insensitive loss function is as follows:

$$\begin{aligned} &\min \quad \frac{1}{2}\| w \|^2 \\ &\text{s.t} \\ &\qquad y_i - < w, x_i > -b \le \varepsilon \\ &\qquad < w, x_i > +b - y_i \le \varepsilon, \end{aligned} \tag{3.37}$$

and ε-insensitive loss function is defined as:

$$Loss\ (y) = \begin{cases} 0 & \text{if} \quad |f(x) - y| \le \varepsilon \\ |f(x) - y| - \varepsilon & \text{otherwise} \end{cases}.$$

The ε-insensitive function allows at most ε deviation between the target and actual values. That is, if we have ε precision, the problem is feasible. However, to allow some errors, the soft margin loss function needs to be considered with slack variables. Therefore, problem (3.37) becomes:

$$\begin{aligned} &\min \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l}(z_i + z_i^*) \\ &\text{s.t} \\ &\qquad y_i - < w, x_i > -b \le \varepsilon + z_i \\ &\qquad < w, x_i > +b - y_i \le \varepsilon + z_i^* \\ &\qquad z_i, z_i^* \ge 0 \end{aligned} \tag{3.38}$$

Minimizing the regularization term $\|w\|^2$ coincides with the flatness of the

function, and the positive real number C plays a role of controlling the amount of penalization for data points lying outside the ε tube. According to the formulation (3.38) and Figure 3.4, one can see the properties of the loss function: The ε tube is fitted to the data, any errors smaller than ε are ignored, and the data points lying outside of the tube are also penalized. We also introduce slack variables to compute the error for underestimating and overestimating the function.

Using the Lagrangian function and optimality conditions, the decision function is obtained as:

$$f(x) = \sum_{i=1}^{l} (\lambda_i - \lambda_i^*) K(x, x_i) + b \tag{3.39}$$

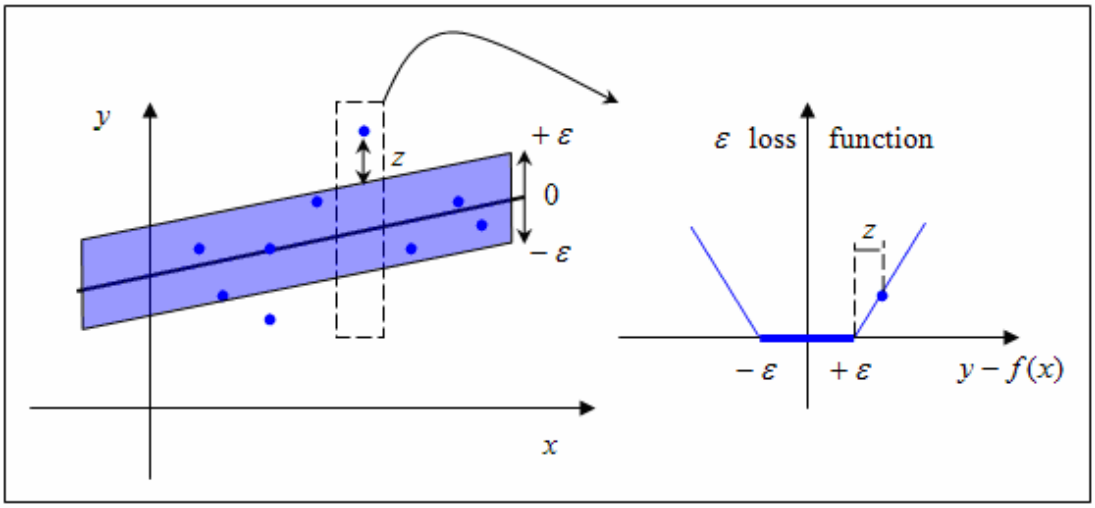

Figure 3. 4 The ε - insensitive loss function for support vector regression. (Schölkopf and Smola, 2002)

where, $K(x, x_i)$ is defined by the kernel function as discussed in chapter 3, and λ_i, λ_i^* are Lagrange multipliers (Vapnik, 1995). Now we deal with the three cases that are discussed in the SVC problem.

Case 1: Perturbation of Data

The soft margin formulation of the regression problem is as follows (Cristianini and Shawe-Taylor, 2000):

$$\begin{aligned} \min \quad & \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l}(z_i + z_i^*) \\ \text{s.t} \quad & \\ & y_i - w^T x_i - b \le \varepsilon + z_i \\ & w^T x_i + b - y_i \le \varepsilon + z_i^* \\ & z_i, z_i^* \ge 0 \end{aligned} \tag{3.40}$$

As shown in the classification cases, we consider the first case of perturbation of data by making a slight change Δx_i in the input data ($x \to x + \Delta x_i$). It is assumed that the perturbations are bounded according to our prior information. The first constraint of the primal formulation (3.40) becomes:

$$\begin{aligned} & y_i - w^T (x_i + \Delta x_i) - b \le \varepsilon + z_i \\ \Rightarrow \quad & y_i - b - w^T x_i - w^T \Delta x_i \le \varepsilon + z_i \\ \Rightarrow \quad & -y_i + b + w^T x_i + w^T \Delta x_i \ge -\varepsilon - z_i \end{aligned} \tag{3.41}$$

We know that constraint (3.41) holds for Δx_i ($\|\Delta x_i\| \le \eta$) in a robust way if and only if the minimum of the left hand side satisfies the following:

$$\min_{\|\Delta x_i\| \le \eta} -y_i + b + w^T x_i + w^T \Delta x_i \ge -\varepsilon - z_i \tag{3.42}$$

Then (3.42) becomes:

$$-y_i + b + w^T x_i + \min_{\|\Delta x_i\| \le \eta} w^T \Delta x_i \ge -\varepsilon - z_i \tag{3.43}$$

By Cauchy-Schwarz inequality, we have $\left|w^T \Delta x_i\right| \le \cdot\|w\| \cdot \|\Delta x_i\| \le \eta \cdot \|w\|$

Therefore, $w^T \Delta x_i$ is bounded as follows:

$$-\eta\|w\| \le w^T \Delta x_i \le \eta\|w\| \tag{3.44}$$

Thus, the minimum of $w^T \Delta x_i$ is equivalent to $-\eta\|w\|$. Then constraint (3.41) becomes:

$$\begin{aligned} & -y_i + b + w^T x_i - \eta\|w\| \ge -\varepsilon - z_i \\ \Rightarrow \quad & y_i - b - w^T x_i + \eta\|w\| \le \varepsilon + z_i \end{aligned} \tag{3.45}$$

The second constraint of the primal formulation (3.40) becomes:

$$\begin{aligned} & w^T (x_i + \Delta x_i) + b - y_i \le \varepsilon + z_i^* \\ \Rightarrow \quad & w^T \Delta x_i \le -w^T x_i + y_i - b + \varepsilon + z_i^* \end{aligned} \tag{3.46}$$

By maximizing $w^T \Delta x_i$ and using of (3.44), the constraint (3.46) becomes:

$$\begin{aligned} \eta\|w\| &\le -w^T x_i + y_i - b + \varepsilon + z_i^* \\ \Rightarrow w^T x_i + \eta\|w\| - y_i + b &\le \varepsilon + z_i^* \end{aligned} \tag{3.47}$$

By (3.45) and (3.47), the primal problem becomes:

$$\begin{aligned} &\min \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l}(z_i + z_i^*) \\ &\text{s.t} \\ &\qquad y_i - w^T x_i - b + \eta\|w\| \le \varepsilon + z_i \\ &\qquad w^T x_i + b - y_i + \eta\|w\| \le \varepsilon + z_i^* \\ &\qquad z_i, z_i^* \ge 0 \end{aligned} \tag{3.48}$$

Case 2: Perturbation of Parameters

In case 2 of perturbation of parameters, we consider a change of the weight vector Δw for problem (3.38) ($w \to w + \Delta w$). The first constraint of (3.38) becomes:

$$\begin{aligned} & y_i - (w + \Delta w)^T x_i - b \le \varepsilon + z_i \\ \Rightarrow\ & y_i - b - w^T x_i - \Delta w^T x_i \le \varepsilon + z_i \\ \Rightarrow\ & -y_i + b + w^T x_i + \Delta w^T x_i \ge -\varepsilon - z_i \end{aligned} \tag{3.49}$$

Note that w is robust feasible with respect to bounded perturbations of the vector w ($\|\Delta w\| \le \eta$), if and only if for every $i = 1, \Lambda\ , l$

$$\min_{\|\Delta w\| \le \eta} -y_i + b + w^T x_i + \Delta w^T x_i \ge -\varepsilon - z_i \tag{3.50}$$

Then (3.50) becomes:

$$-y_i + b + w^T x_i + \min_{\|\Delta w\| \le \eta} \Delta w^T x_i \ge -\varepsilon - z_i \tag{3.51}$$

By Cauchy-Schwarz inequality, again we have:

$$\left|\Delta w^T x_i\right| \le \|\Delta w\| \cdot \|x_i\| \le \eta \cdot R. \tag{3.52}$$

Note we assume that $\|x_i\| \le R$. Then $\Delta w^T x_i$ is bounded as follows:

$$-\eta R \le \Delta w^T x_i \le \eta R \tag{3.53}$$

Therefore, the minimum value of $\Delta w^T x_i$ is equal to $-\eta R$. Then constraint (3.51) becomes:

$$\begin{aligned} & -y_i + b + w^T x_i - \eta R \ge -\varepsilon - z_i \\ \Rightarrow \quad & y_i - b - w^T x_i + \eta R \le \varepsilon + z_i \end{aligned} \tag{3.54}$$

The second constraint of (3.38) becomes:

$$\begin{aligned} & w^T x_i + b - y_i \le \varepsilon + z_i^* \\ \Rightarrow & (w + \Delta w)^T x_i + b - y_i \le \varepsilon + z_i^* \\ \Rightarrow & \Delta w^T x_i \le \varepsilon + z_i^* \\ \Rightarrow & \Delta w^T x_i \le -w^T x_i - b + y_i + \varepsilon + z_i^* \end{aligned} \tag{3.55}$$

By maximizing $\Delta w^T x_i$ and using of (3.52), the constraint (3.55) becomes:

$$\begin{aligned} & \eta R \le -w^T x_i - b + y_i + \varepsilon + z_i^* \\ \Rightarrow & w^T x_i + \eta R + b - y_i \le \varepsilon + z_i^* \end{aligned} \tag{3.56}$$

By (3.54) and (3.56), hence the primal problem becomes:

$$\min \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l}(z_i + z_i^*)$$

s.t

$$\begin{aligned} & y_i - w^T x_i - b + \eta R \le \varepsilon + z_i \\ & w^T x_i + b - y_i + \eta R \le \varepsilon + z_i^* \\ & z_i, z_i^* \ge 0 \end{aligned} \tag{3.57}$$

Case 3: Perturbation of Input Data and Parameter

In this case, we consider perturbation of data and parameters simultaneously ($x \to x + \Delta x_i, w \to w + \Delta w$).

The first constraint of (3.38) becomes:

$$\begin{aligned} & y_i - b - (w^T x_i + w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)) \le \varepsilon + z_i \\ \Rightarrow\ & y_i - b - (w^T x_i + w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)) \le \varepsilon + z_i \\ \Rightarrow\ & y_i - b - w^T x_i - (w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)) \le \varepsilon + z_i \\ \Rightarrow\ & -y_i + b + w^T x_i + (w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)) \ge -\varepsilon - z_i \end{aligned} \tag{3.58}$$

We know that the constraint of (3.58) holds for Δx and Δw, if and only if

$$\begin{aligned} & \min_{\|\Delta x_i\| \le \eta_1, \|\Delta w\| \le \eta_2} \ -y_i + b + w^T x_i + (w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)) \ge -\varepsilon - z_i \\ \Rightarrow\ & -y_i + b + w^T x_i + \min_{\|\Delta x_i\| \le \eta_1, \|\Delta w\| \le \eta_2} (w^T \Delta x_i + \Delta w^T x_i + (\Delta w)^T (\Delta x_i)) \ge -\varepsilon - z_i \end{aligned} \tag{3.59}$$

By Cauchy-Schwarz inequality, we have:

$$\begin{aligned} & -\eta_1 \|w\| \le w^T \Delta x_i \le \eta_1 \|w\| \\ & -\eta_2 R \le \Delta w^T x_i \le \eta_2 R \end{aligned}, \tag{3.60}$$

where we assume $\|x_i\| \le R$.

Similarly by Cauchy-Schwarz, we have $\left|(\Delta w)^T(\Delta x_i)\right| \le \|\Delta w\| \cdot \|\Delta x_i\| \le \eta_1 \cdot \eta_2$.

Hence,

$$-\eta_1\|w\| - \eta_2 R - \eta_1\eta_2 \le \left|[w^T\Delta x_i + \Delta w^T x_i + (\Delta w)^T(\Delta x_i)]\right| \le \eta_1\|w\| + \eta_2 R + \eta_1\eta_2 \qquad (3.61)$$

Therefore,

$$\min_{\|\Delta x_i\|\le\eta_1, \|\Delta w\|\le\eta_2} [w^T\Delta x_i + \Delta w^T x_i + (\Delta w)^T(\Delta x_i)] = -\eta_1\|w\| - \eta_2 R - \eta_1\eta_2 \qquad (3.62)$$

The minimum value of $w^T\Delta x_i + \Delta w^T x_i + (\Delta w)^T(\Delta x_i)$ is equivalent to $-\eta_1\|w\| - \eta_2 R - \eta_1\eta_2$. Then constraint (3.55) becomes:

$$\begin{aligned} & -y_i + b + w^T x_i - \eta_1\|w\| - \eta_2 R - \eta_1\eta_2 \ge -\varepsilon - z_i \\ \Rightarrow \quad & y_i - b - w^T x_i + \eta_1\|w\| + \eta_2 R + \eta_1\eta_2 \le \varepsilon + z_i \end{aligned} \qquad (3.63)$$

The second constraint of (3.38) becomes:

$$\begin{aligned} & (w^T x_i + w^T\Delta x_i + \Delta w^T x_i + (\Delta w)^T(\Delta x_i)) + b - y_i \le \varepsilon + z_i^* \\ \Rightarrow\ & (w^T\Delta x_i + \Delta w^T x_i + (\Delta w)^T(\Delta x_i)) \le -w^T x_i - b + y_i + \varepsilon + z_i^* \end{aligned} \qquad (3.64)$$

By maximizing $w^T\Delta x_i + \Delta w^T x_i + (\Delta w)^T(\Delta x_i)$ and the result of (3.61), the constraint (3.64) becomes:

$$\begin{aligned} & \eta_1\|w\| + \eta_2 R + \eta_1\eta_2 \le -w^T x_i - b + y_i + \varepsilon + z_i^* \\ \Rightarrow\ & \eta_1\|w\| + \eta_2 R + \eta_1\eta_2 + w^T x_i + b - y_i \le \varepsilon + z_i^* \end{aligned} \qquad (3.65)$$

Finally by (3.63) and (3.65), the maximal margin classifier problem considering

uncertainty will be as follows:

$$\min \quad \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l}(z_i + z_i^*)$$

s.t

$$\begin{aligned} & y_i - w^T x_i - b + \eta_1\|w\| + \eta_2 R + \eta_1\eta_2 \le \varepsilon + z_i \\ & w^T x_i + b - y_i + \eta_1\|w\| + \eta_2 R + \eta_1\eta_2 \le \varepsilon + z_i^* \\ & z_i, z_i^* \ge 0 \end{aligned} \tag{3.66}$$

CHAPTER 4. Applications and Numerical Examples

4.1 Computational Results for Sensitivity and Robust Analysis applied to SVC

To illustrate the analysis of 3.2.1, we consider two well known linearly separable and nonlinearly separable examples to the support vector classification (SVC) problem. We assume that we have 4 input data points (1, 1), (1, -1), (-1, 1) and (-1, -1) with labels +1 and -1 respectively. All the experiments for this example have been performed by MATLAB and TOMLAB/SNOPT toolbox.

Table 4. 1 Relations between input and output of AND / XOR Problem.

X_1	X_2	AND	XOR
1	1	1	1
1	-1	-1	-1
-1	1	-1	-1
-1	-1	-1	1

4.1.1 Linearly Separable Case (AND Problem)

Case 1: Perturbation of Data

Formulation (3.26) is the SVC problem with respect to the perturbation of input data. By changing the data perturbations we are able to inspect the sensitivity of the classification problem. We can solve the AND classification problem with different values of η as showing in Table 4.2. As expected, the results have been quite interesting. First, the separating line does not change even if the uncertainty parameter η is changed. In the precise case (η=0), for example, the separating line becomes

$$w^T x + b = 0 \Rightarrow \begin{pmatrix} w_1 & w_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + b = (1 \quad 1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - 1 = x_1 + x_2 - 1 = 0 .$$

Table 4.2 shows the output of the support vector classification for the AND problem using perturbations of data by varying the uncertainty parameter η. Note that the separating line is exactly the same line as shown in Figure 4.1 for several values of η. The other interesting outcome is related to the margin of separation. Note that if we increase the uncertainty level, then the margin of separation becomes smaller. Observe that the maximum perturbation value of η for which the separation line does not

change is $\sqrt{2}/2$. This says that the SVM solution is robust.

Table 4. 2 SVC Output using perturbations of input data for AND problem

Eta (η)	w_1	w_2	b	margin
0	1.0000	1.0000	-1.0000	0.7071
0.1	1.1647	1.1647	- 1.1647	0.6552
0.2	1.3944	1.3944	- 1.3944	0.5988
0.3	1.7369	1.7369	- 1.7369	0.5365
0.4	2.3025	2.3025	- 2.3025	0.4660

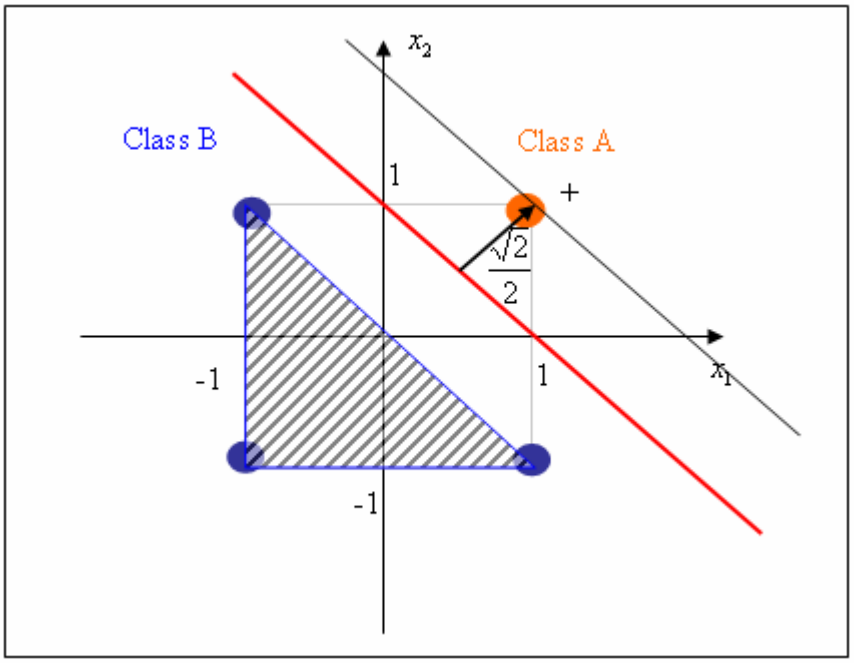

Figure 4. 1 Illustration of the hyperplane and margin using the AND problem

Case 2: Perturbation of Parameter

Equation (3.31) shows a primal formulation for the SVC problem with perturbations of the weight parameter w. In this case, we assume the data are bounded by the maximum radius *R* of length 1. The separation line is not changed similar to case 1. If the uncertainty parameter η becomes 0, the problem represents the precise case and the result is as in Figure 4.1. If we increase the uncertainty parameter η, the separating margin is decreasing. It is similar to the result of case 1. Case 1 (perturbation of input), however, is more sensitive than case 2 (perturbation of parameter).

Table 4. 3 SVC Output using perturbation of parameter for the AND problem

Eta (η)	w_1	w_2	b	margin
0	1.0000	1.0000	-1.0000	0.7071
0.1	1.1000	1.1000	-1.1000	0.6428
0.2	1.2000	1.2000	-1.2000	0.5893
0.3	1.3000	1.3000	-1.3000	0.5439
0.6	1.6000	1.6000	-1.6000	0.4419
1.0	2.0000	2.0000	-2.0000	0.3536

In addition, the result of case 2 is also influenced by the radius of the sphere bounding the data. If the maximum radius *R* decreases, the separating margin also will be decreased. Table 4.3 shows the output of the support vector classification problem using perturbations of parameters by varying the uncertainty parameter η.

Case 3: Perturbation of Parameters and Data

In case 3, we examine the perturbation of parameter w and the data at the same time. Here, we use two bounded values η_1 and η_2 for each parameter and data. The formulation of case 3 shows more complicated constraints compared to case 2 and case 3. This is mostly due to the fact that case 3 is created with a combination of parameters and data unlike case 1 and case 2. Table 4.4 shows the output of the support vector classification problem using perturbations of data and parameters respectively. Here we experiment with a fixed η_1 and varying η_2 for several values of these parameters. If we consider η_2=0, the problem becomes case 1. On the other hand, it will be case 2 when η_1 becomes 0.

To compare the impact of the uncertainty parameter, we experiment with

several value of η_1, η_2 such as $\eta_1 = 0.1$, $\eta_2 = 0.3$ and $\eta_1 = 0.3$, $\eta_2 = 0.1$ as shown in Table 3.4. In this table, we observe that the second case ($\eta_1 = 0.3$, $\eta_2 = 0.1$) has a smaller margin than in the first case. This means that the uncertainty parameter of η_1 input has more sensitivity effects than the uncertainty parameter of η_2. This table also demonstrates the complexity of the combined affects of the elements.

Table 4. 4 SVC Output using perturbation of data and parameter for AND problem.

Eta 1(η_1)	Eta 2(η_2)	w_1	w_2	b	margin
0	0	1.0000	1.0000	-1.0000	0.7071
	0.1	1.1000	1.1000	-1.1000	0.6428
	0.3	1.3000	1.3000	- 1.3000	0.5439
	0.5	1.5000	1.5000	-1.5000	0.4714
0.1	0	1.1647	1.1647	-1.1647	0.6071
	0.1	1.2928	1.2928	-1.2928	0.5469
	0.3	1.5491	1.5491	-1.5491	0.4565
	0.5	1.8053	1.8053	-1.8053	0.3917
0.3	0	1.7369	1.7369	-1.7369	0.4071
	0.1	1.9627	1.9627	-1.9627	0.3603
	0.5	2.8659	2.8659	-2.8659	0.2467

4.1.2 Linearly Non-separable Case (Exclusive OR (XOR) Problem)

In the previous Chapter the support vector classification (SVC) problem was applied to the linearly separable case. Here we use a polynomial kernel function with degree 2 to extend our approach on the nonlinearly separable case. The polynomial kernel function is formulated as $K(x_i, x_j) = (x_i^T x_j + 1)^2$. Using the data in Table 4.1, the kernel matrix can be computed as follows:

$$K = \begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & K(x_1,x_3) & K(x_1,x_4) \\ K(x_2,x_1) & K(x_2,x_2) & K(x_2,x_3) & K(x_2,x_4) \\ K(x_3,x_1) & K(x_3,x_2) & K(x_3,x_3) & K(x_3,x_4) \\ K(x_4,x_1) & K(x_4,x_2) & K(x_4,x_3) & K(x_4,x_4) \end{bmatrix} = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix} \quad (4.1)$$

In section 3.1.2 we have derived a nonlinearly separable problem formulation. Perturbation of input data (Case 1) is illustrated for the XOR problem. Equation (3.26) can be handled by kernelization and it is also known that

$$w = \sum_{i=1}^{l} y_i \alpha_i \varphi(x_i), \quad (4.2)$$

where α_i is non-negative Lagrange multiplier and y_i is the label. Now substitute equation (4.2) to problem (3.26), then the objective function of the problem becomes:

$$\frac{1}{2}\| w \|^2 = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i \alpha_j y_i y_j (x_i \cdot x_j) = \frac{1}{2}\alpha^T \tilde{K}\alpha \tag{4.3}$$

where, $\tilde{K} = \begin{bmatrix} 9 & -1 & -1 & 1 \\ -1 & 9 & 1 & -1 \\ -1 & 1 & 9 & -1 \\ 1 & -1 & -1 & 9 \end{bmatrix}$.

Using equation (4.3), the norm of *w* can be represented by as follows:

$$\|w\| = \sqrt{\alpha^T \tilde{K}\alpha} \tag{4.4}$$

By substituting of (4.3) and (4.4) to equation (3.26), the robust formulation will be as follows:

$$\begin{aligned} \min_{\alpha,b,t} \quad & \frac{1}{2}\alpha^T \tilde{K}\alpha + C\sum_{i=1}^{l} t_i \\ s.t \quad & \tilde{K}_i\alpha + y_i b - \eta\sqrt{\alpha^T \tilde{K}\alpha} \geq 1 - t_i \\ & \alpha_i \geq 0, \qquad i = 1,\ldots,l. \end{aligned} \tag{4.5}$$

The formulation (4.5) is a constrained nonlinear optimization problem. The solutions of the XOR problem with different uncertainty parameters η are summarized in Table 4.5. The experiment increases the uncertainty parameter η from 0 to a value that gives infeasibility. From the results, we know that we can obtain the separating solution when the level of uncertainty does not exceed 1.42.

As expected, if we increase the uncertainty parameter η, the value of α_i is increased and the problem is still separable. In the precise case ($\eta = 0$), we obtain the

following results:

$$\alpha = \begin{pmatrix} 0.125 & 0.125 & 0.125 & 0.125 \end{pmatrix} \tag{4.6}$$

We can compute the discriminant function, which is given by:

$$f(x) = \sum_{i=1}^{4} y_i \alpha_i K(x, x_i) + b = x_1 x_2 \tag{4.7}$$

It is well known that the margin of the precise case is $\sqrt{2}$. That means we can make perturbation of input data up to $\sqrt{2}$. Table 4.5 explains the relationship between the margin and uncertainty level.

Table 4. 5 SVC Output using perturbations of input data for XOR problem

Eta (η)	α_1	α_2	α_3	α_4	b
0	0.1250	0.1250	0.1250	0.1250	0.0000
0.3	0.1587	0.1587	0.1587	0.1587	0.0000
0.6	0.2171	0.2171	0.2171	0.2171	0.0000
0.9	0.3438	0.3438	0.3438	0.3438	0.0000
1.2	0.8252	0.8252	0.8252	0.8252	0.0000
1.4	12.4372	12.4372	12.4372	12.4372	0.0000
1.41	41.9542	41.9542	41.9542	41.9542	0.0000
1.42	No feasible solution				

4.1.3 Implementation on Real Data

From the AND and XOR problem examples, we obtain the fact that the uncertainty of input has more sensitivity effects than the uncertainty of parameter. To carry out the implementation, the well known Breast Cancer Wisconsin data is considered (Mangasarian and Wolberg), where the two classes (malignant and benign) could be decided from 9 attributes of the patients (683 data points are used). For the experiment, 50% of the data points are used for training and the rest are placed in the testing data set.

The experiments are controlled by varying the perturbation parameter η for the three cases. The margin of the experiments is summarized in Table 4.6 and Table 4.7. Table 4.6 and Figure 4.2 show how much the variation of the input data and parameter can influence the output of the learning machine. We also provide Table 4.7, for case 3, which has considered the perturbation of both input and parameters. We found that if the perturbation of input or parameter is increased, the margin between the two classes decreases.

In order to visualize the sensitivity of the margin between Case 1 and Case 2,

we provide a comparison of margin sensitivity in Figure 4.2 and Figure 4.3. The margin of Case 1 rapidly decreases as the perturbation parameter η increases, but the margin of Case 2 gradually decreases compared to Case 1. In Figure 4.3 the gap of margin is slightly decreased as the parameters' perturbation level decreased.

The margin is decreased with both perturbations of data and parameters and it is increased when one of the perturbations is not affected: This shows the largest margin when the perturbation of parameters is zero. Therefore, the real data experiment also shows a similar result as in the simple cases experiments.

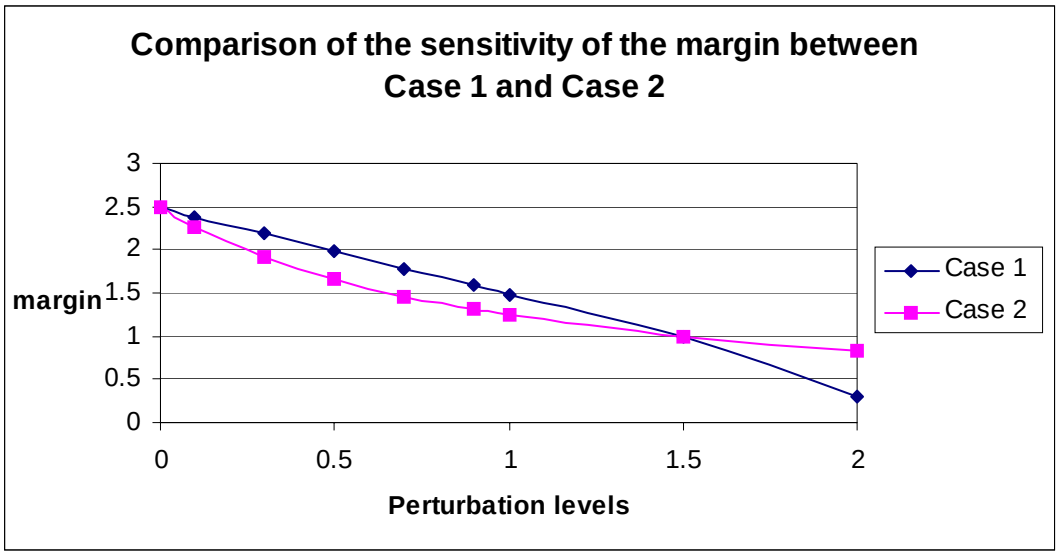

Figure 4. 2 Comparison of the margin between the two cases. Case 1 looks more sensitive than Case 2.

Table 4. 6 Comparison of SVC Output using perturbation of data (Case 1) and

parameter (Case 2) for Breast Cancer Wisconsin data.

eta	0	0.1	0.3	0.5	0.7	0.9	1	1.5	2
Case 1	2.4808	2.3808	2.1808	1.9808	1.7808	1.5808	1.4808	0.9808	0.3075
Case 2	2.4808	2.2553	1.9083	1.6539	1.4593	1.3057	1.2404	0.9923	0.8269

Table 4. 7 SVC Output using perturbation of data and parameter for Breast Cancer Wisconsin data.

eta 1(η_1)	eta 2(η_2)	margin	eta 1(η_1)	eta 2(η_2)	margin
0	0	2.4808	0.5	0	1.9808
	0.1	2.2553		0.1	1.8007
	0.5	1.6539		0.5	1.3205
	1.0	1.2404		1.0	0.9904
0.1	0	2.3808	1.0	0	1.4808
	0.1	2.1644		0.1	1.3462
	0.5	1.5872		0.5	0.9872
	1.0	1.1904		1.0	0.7404

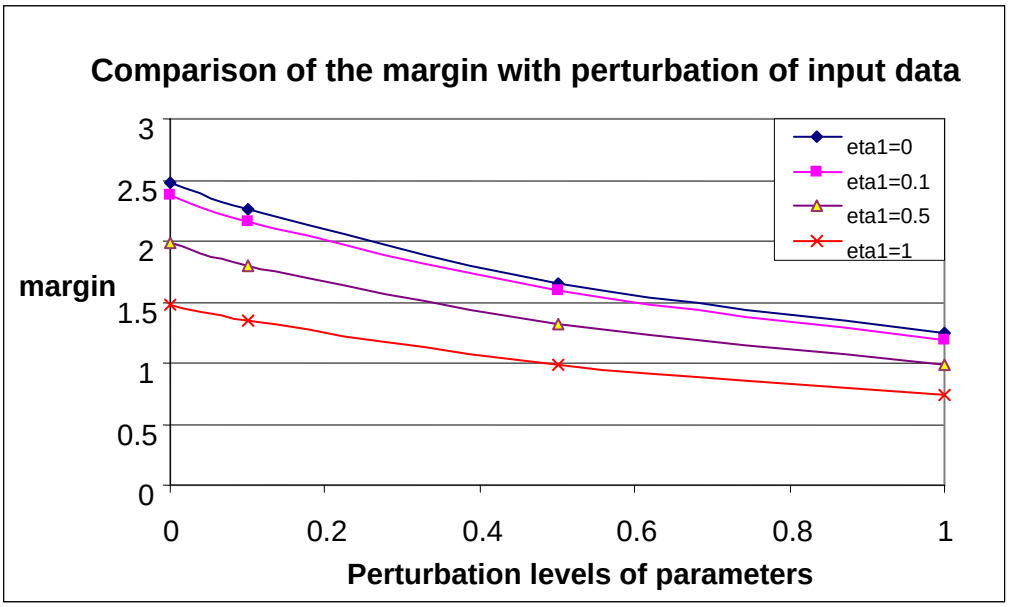

Figure 4. 3 Comparison of the margin between the perturbation of input data and parameters.

4.2 Computational Results for Sensitivity and Robust Analysis Applied to SVR

4.2.1 Traffic Data Analysis

From the basic concepts of perturbation of input data and parameters, robust SVR is applied to real time traffic data. The traffic data are provided by the Freeway Performance Measurement System (PeMS) based on varying speeds recorded by traffic sensors. To predict exact vehicle speeds at the specific spot, the data were

collected in California by sensor 761552 during the year 2002. For the purpose of predicting regular weekday vehicle speeds, we only used Monday data except holiday and some special event days.

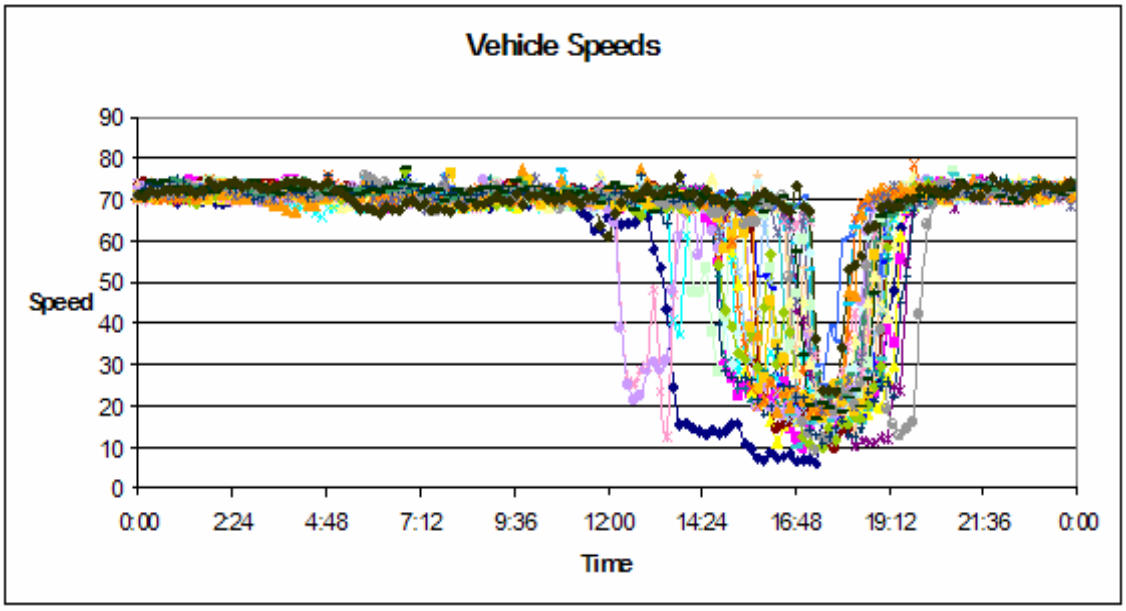

Figure 4. 4 Freeway vehicle speed patterns during Monday 2002. There shows traffic congestion between 3 pm and 8 pm.

We used data from the first 20 weeks for the training set and used the last 10 weeks as the testing set. The speed measurements were taken at 10 minutes intervals for the whole day. Figure 4.3 shows vehicle speeds on a daily basis for all Mondays during 2002. In the experiment, polynomial kernel function with degree 2

outperformed RBF and sigmoid kernel functions, even if RBF and sigmoid kernel function also performed well. The SVR experiments were performed in the MATLAB and TOMLAB/SNOPT toolbox (Holmström, 1999).

Case 1: Perturbation of Input Data

In the previous chapter the support vector regression (SVR) problem with perturbed data and parameters was constructed. Problem (3.48) can be expressed as a nonlinear regression problem by means of the kernel method. When we map the data, the weight vector can be represented as follows:

$$w = \sum_{i=1}^{l} \alpha_i \, \varphi(x_i) \tag{4.8}$$

The inner product can be replaced as kernel function, such as $K(x, x_i) = < \varphi(x_1), \varphi(x_2) >$. Then the weight vector w can be represented as kernel function,

$$\|w\| = \sqrt{\alpha^T K \alpha}. \tag{4.9}$$

By substituting (4.8), (4.9) into the robust SVR problem, we have the following formulation:

$$
\begin{aligned}
\min \quad & \frac{1}{2}\alpha^T K\ \alpha + C\sum_{i=1}^{l}(z_i + z_i^*) \\
\text{s.t} \quad & y_i - K_i\alpha - b + \eta\sqrt{\alpha^T K\,\alpha} \le \varepsilon + z_i \\
& K_i\alpha + b - y_i + \eta\sqrt{\alpha^T K\,\alpha} \le \varepsilon + z_i^* \\
& z_i, z_i^* \ge 0.
\end{aligned}
\tag{4.10}
$$

Problem (4.10) is a constrained nonlinear optimization problem.

To investigate the behavior of the SVR solution with perturbed data, we varied the uncertainty parameter η with different levels from 0 to 5. Figure 4.3 illustrates the vehicle speed with time-varying condition. Figure 4.4 and Table 4.8 show the fact that the relative absolute error (RAE) increases as the perturbation level increased. From the results, we know that we can obtain the accurate solution when the level of uncertainty decreased.

Table 4. 8 Relative Absolute Error (RAE) results for the five different data perturbation levels.

$\eta = 0$	$\eta = 1$	$\eta = 2$	$\eta = 3$	$\eta = 4$	$\eta = 5$
0.066093	0.080778	0.140899	0.19447	0.25047	0.284357

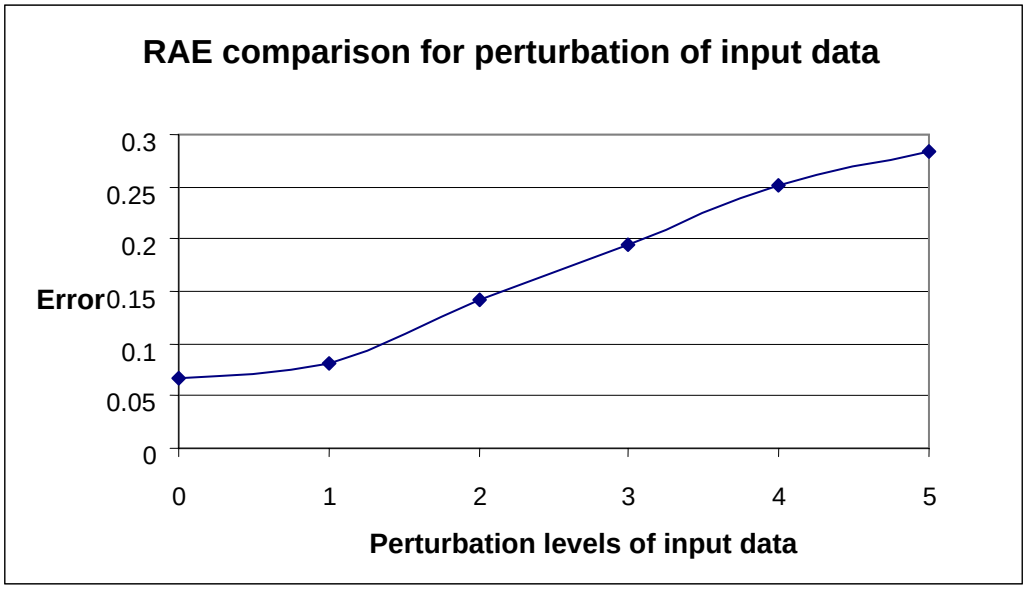

Figure 4. 5 Relative Absolute Error (RAE) with different perturbation level η. It is natural result that the RAE increases as η increased.

Case 2: Perturbation of Parameter

Similar to the previous case, problem (3.57) can be reformulated as follows;

$$\begin{aligned} \min \quad & \frac{1}{2}\alpha^T K\ \alpha + C\sum_{i=1}^{l}(z_i + z_i^*) \\ \text{s.t} \quad & y_i - K_i\alpha - b + \eta R \le \varepsilon + z_i \\ & K_i\alpha + b - y_i + \eta R \le \varepsilon + z_i^* \\ & z_i, z_i^* \ge 0. \end{aligned} \tag{4.11}$$

In the robust SVR formulation (4.11), we assume that the radius of the data is known by prior information. The experiments are controlled by varying the perturbation

parameter η and the radius of data *R*.

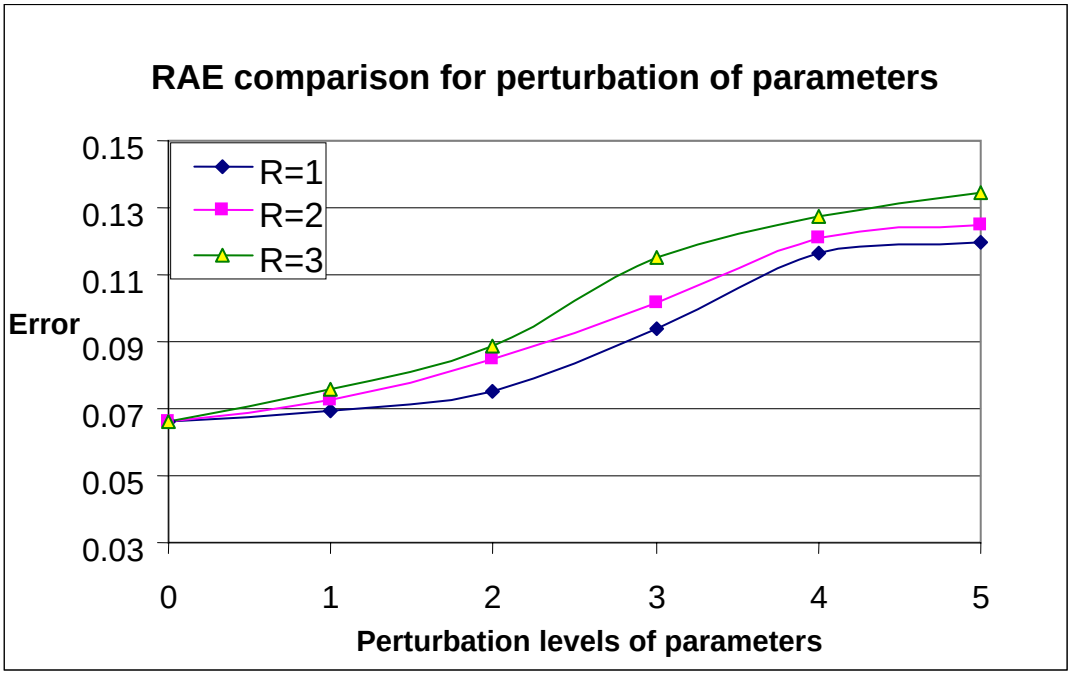

Figure 4. 6 Comparison of RAE. Three cases of experiment for different bounded radius of data are illustrated. A large number of η and *R* decrease the prediction accuracy.

Table 4. 9 Relative Absolute Error (RAE) results for the five different parameter perturbation levels.

	$\eta = 0$	$\eta = 1$	$\eta = 2$	$\eta = 3$	$\eta = 4$	$\eta = 5$
R=1	0.066093	0.069518	0.074953	0.094155	0.116601	0.119611
R=2	0.066093	0.072552	0.084728	0.101773	0.120689	0.124552
R=3	0.066093	0.075553	0.088393	0.114934	0.127619	0.134553

Figure 4.6 and Table 4.9 display three cases of bounded data radius (R=1, 2, 3) for the different parameters' perturbation level. The RAE increased when the perturbation level and radius of data are increased. The slopes of the errors, however, are not steeper than case 1. The gap of the errors among the three cases of R is relatively stable as the perturbation level increased.

Case 3: Perturbation of Input Data and Parameters

Now we have a formulation considering both the perturbation of input data and parameters.

$$\begin{aligned} \min \quad & \frac{1}{2}\alpha^T K\ \alpha + C\sum_{i=1}^{l}(z_i + z_i^*) \\ \text{s.t} \quad & y_i - K_i\alpha - b + \eta_1\sqrt{\alpha^T K\,\alpha} + \eta_2 R + \eta_1\eta_2 \le \varepsilon + z_i \\ & K_i\alpha + b - y_i + \eta_1\sqrt{\alpha^T K\,\alpha} + \eta_2 R + \eta_1\eta_2 \le \varepsilon + z_i^* \\ & z_i, z_i^* \ge 0. \end{aligned} \tag{4.12}$$

The uncertainty levels η_1 and η_2 are related to the input data perturbation and parameters' perturbation, respectively. Figure 4.6 and Figure 4.7 illustrate relations between the different perturbation levels of input data and parameters.

The two figures show 36 combinations (6 data perturbation levels and 6 parameters' perturbation levels) between the two different uncertainty levels η_1 and η_2 (see

Table 4.10). Figure 4.6 shows the changes of error based on the input data perturbation level η_1, on the other hand, Figure 4.7 illustrates the changes of error based on the parameters' perturbation level η_2. Figure 4.6 and Figure 4.7 show absolutely different shapes and slopes.

Table 4.10 Combinational errors between the perturbation of input data and parameters.

	$\eta_2 = 0$	$\eta_2 = 1$	$\eta_2 = 2$	$\eta_2 = 3$	$\eta_2 = 4$	$\eta_2 = 5$
$\eta_1 = 0$	0.066093	0.069518	0.074953	0.094155	0.116601	0.119611
$\eta_1 = 1$	0.080778	0.082114	0.093068	0.112414	0.120808	0.122936
$\eta_1 = 0$	0.140899	0.124219	0.124301	0.137284	0.123594	0.137069
$\eta_1 = 0$	0.19447	0.190648	0.168811	0.164644	0.162148	0.173636
$\eta_1 = 0$	0.25047	0.21064	0.21064	0.184649	0.186852	0.194428
$\eta_1 = 0$	0.284357	0.281115	0.247831	0.195007	0.210073	0.206718

From Figure 4.6 we find a pattern that the RAE increases as the input perturbation level increased. It is similar to Figure 4.4, which just considered perturbation of input data. Therefore, it is found that the perturbation of input data affects the solutions of the optimization problem more, compared with the perturbation of parameters.

From Figure 4.7 it is difficult to conclude that a pattern exists. The RAE

increases when the data perturbation level η_1 is small (η_1=0 and 1), however, it decreases when the data perturbation level η_1 is large (η_1=4 and 5). The error gap also decreases when the parameters' perturbation level η_2 increases. The error gap is extremely large when the parameters' perturbation level η_2 is zero. Therefore, we conclude that the perturbation of input data is more significant than the perturbation of parameters to the solution. This observation is the same as in the SVC problem as shown in the previous section.

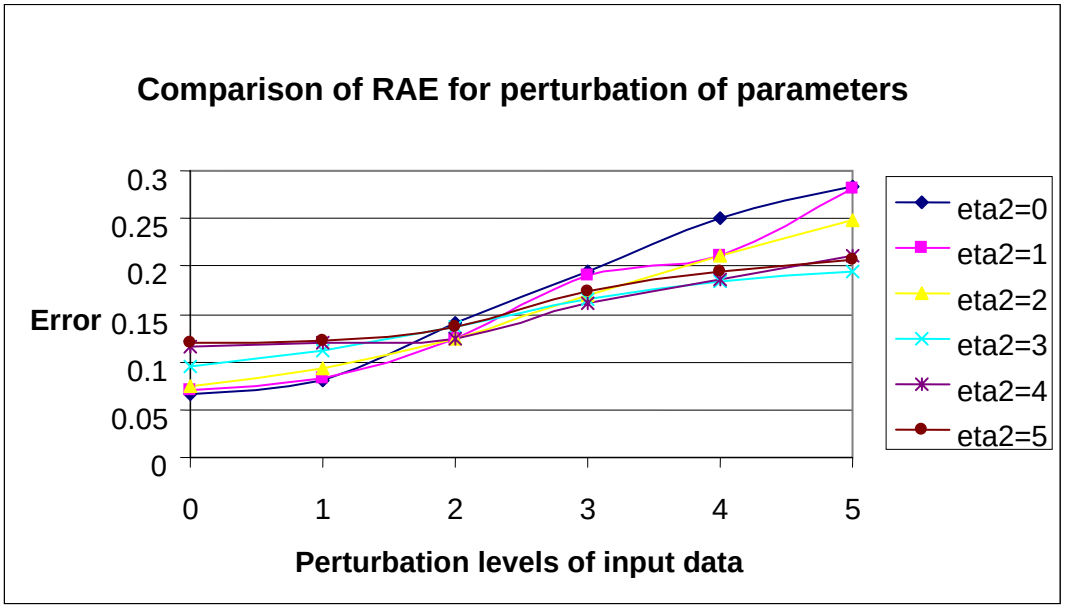

Figure 4. 7 Compare RAE for parameter perturbation based on the input data perturbation level η_1.

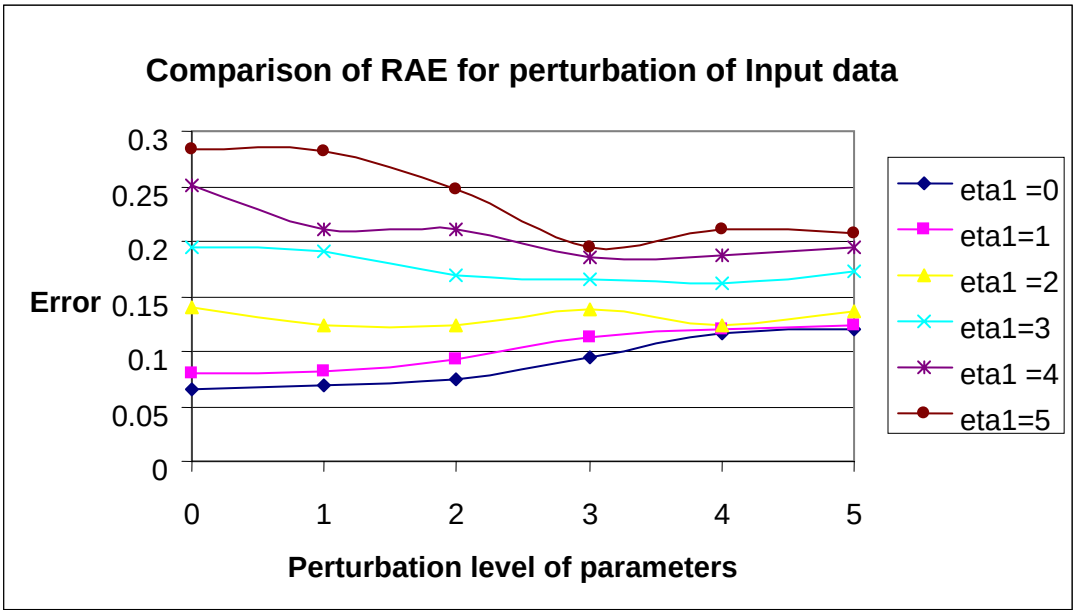

Figure 4. 8 Compare RAE for input data perturbation based on the parameter perturbation level η_2

By examining the three cases of sensitivity analysis, we found facts that:

- Perturbation of input is more sensitive than the perturbation of parameters
- However, if we consider the perturbation of parameters at the same time, the effects of perturbation of input is decreased by the counter effect between the two perturbations.

CHAPTER 5. Support Vector Machines Using Uncertain Programming Approach

5.1 Probability Constrained Approach

When it comes to considering an uncertainty framework for the optimization model problem, one generally looks into either robust optimization or probability constrained optimization. While the robust optimization approach handles the uncertainty based on an uncertainty set, probability constrained optimization takes into account the probability distribution for the uncertainty. In this chapter, we investigate a novel probability constrained approach and scenario constrained approach as well.

As discussed in chapter 2, several researchers have investigated robust optimization problems with different shapes of bounded uncertainty sets. Since the robust optimization problem becomes an SOCP problem, which is difficult to solve, researchers Calafiore and Campi (2005) tried to use sampling schemes to represent bounded uncertain sets through random constraints. However, the suggested approaches have a drawback. The number of random constraints is increasing with the

accuracy of the optimal solution resulting in a large scale optimization problem. Therefore, we provide a new alternative robust optimization problem.

The probability constrained approach described in this chapter will take into consideration a bounded knowledge set in the neighborhood of a training point x_i where each point in this knowledge set keeps the same label value y_i. Replicated observations are randomly selected in the knowledge set, rather than working with well-defined data points. Suppose we are given a training data set x_i and labels y_i, $i = 1, \Lambda, l$, where $x_i \in R^n$ and y_i belongs to positive (+1) or negative (-1) classes. Replicated data points $x_{i_1}, \Lambda, x_{i_N}$ which belong to the knowledge set $V(x_i)$, are also considered.

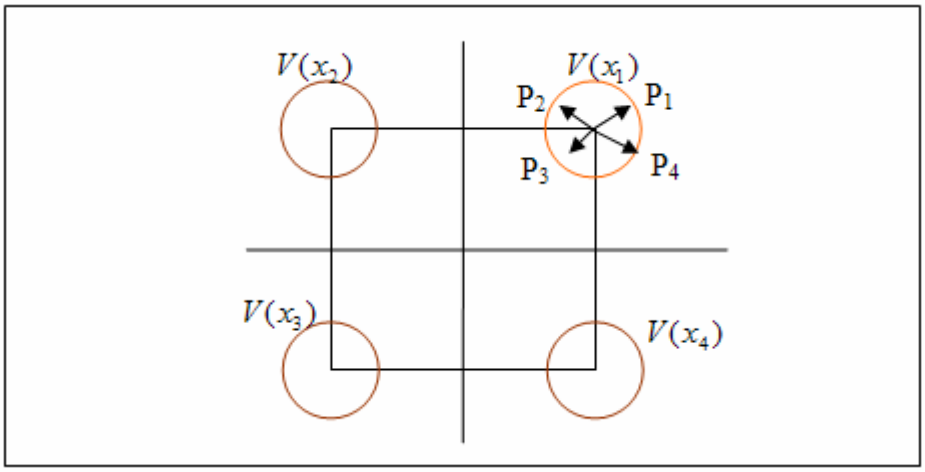

Figure 5. 1 Illustration of four Knowledge sets ($V(x_i)$) and four replicated data points (p_i) in the knowledge sets.

Figure 5.1 illustrates the concept of knowledge sets $V(x_i)$ and replicated measurements p_i ; we have four knowledge or vicinity sets and four replicated measurements in each knowledge set. It is assumed that each replicated measurement has its own probability, and the sum of probabilities within an uncertainty set is one. If we consider equal probability for each replication, the primal optimization problem can be written as:

$$\begin{aligned} &\min_{w,b,t} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} t_i \\ &s.t \quad y_i[\frac{1}{N}\sum_{j=1}^{N} < w, x_{ij} > +b] \geq 1 - t_i \ , \\ &\qquad i = 1, \Lambda \ , l \end{aligned} \tag{5.1}$$

where i is the number of training points (number of knowledge sets), and j is the number of replications within each knowledge set of a training point. In the general case, the generalized constraints of problem (5.1) can be expressed as follows:

$$y_i[\sum_{j=1}^{N} P_{ij} < w, x_{ij} > +b] \geq 1 - t_i, \ \ i = 1, \Lambda \ , l \ . \tag{5.2}$$

P_{ij} in the constraint (5.2) represents a probability for each replicated measurement j in the given knowledge set i. The selection of distribution functions for the probabilities in (5.2) influences the SVM solutions. By considering probability constraint in problem (5.1), we can employ the following probability constrained optimization

problem.

$$\begin{aligned} &\min_{w,b,t} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} t_i \\ &s.t \ \ y_i[\sum_{j=1}^{N} P_{ij} < w, x_{ij} > +b] \geq 1 - t_i \\ &\qquad t_i \geq 0, \qquad\qquad i = 1, \Lambda\ , l \end{aligned} \tag{5.3}$$

Problem (5.3) can be transformed into an unconstrained optimization problem by introducing Lagrange multipliers.

$$L(w,b,\alpha,\beta) = \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l} t_i - \sum_{i=1}^{l} \alpha_i [y_i (\sum_{j=1}^{N} P_{ij} < w, x_{ij} > +b) - 1 + t_i] - \sum_{i=1}^{l} \beta_i t_i \ , \tag{5.4}$$

where α , β are the Lagrange multipliers. By the optimality conditions, the partial derivative of weight vector *w* must satisfy the following conditions:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} \alpha_i y_i \sum_{j=1}^{N} P_{ij} x_{ij} = 0 \tag{5.5}$$

From the above conditions, we derive:

$$w = \sum_{i=1}^{l} \alpha_i y_i \sum_{j=1}^{N} P_{ij} x_{ij} \ , \tag{5.6}$$

which has a similar form to the traditional SVM solution. Substituting (5.6) into the objective function of problem (5.3), we have:

$$
\begin{aligned}
\frac{1}{2} < w, w >= \frac{1}{2} < \sum_{i=1}^{l} \alpha_i y_i \sum_{j=1}^{N} P_{ij} x_{ij}, \sum_{k=1}^{l} \alpha_k y_k \sum_{d=1}^{N} P_{kd} x_{kd} > \\
= \frac{1}{2} \sum_{i=1}^{l} \sum_{k=1}^{l} \alpha_i \alpha_k y_i y_k \sum_{j=1}^{N} \sum_{d=1}^{N} P_{ij} P_{kd} < x_{ij}, x_{kd} > \\
= \frac{1}{2} \sum_{i=1}^{l} \sum_{k=1}^{l} \alpha_i \alpha_k y_i y_k < \sum_{j=1}^{N} P_{ij} x_{ij}, \sum_{d=1}^{N} P_{kd} x_{kd} > \\
= \frac{1}{2} \alpha^T \tilde{K} \alpha,
\end{aligned}
\tag{5.7}
$$

where $\tilde{k}_{ik} = y_i y_k < \sum_{j=1}^{N} P_{ij} x_{ij}, \sum_{d=1}^{N} P_{kd} x_{kd} >$, i and k are the number of samples, and j and d are the number of replicated observations.

The constraint of (5.3) also can be simplified by the use of modified kernel function $\tilde{K}$. The constraint can be derived as follows:

$$
\begin{aligned}
& y_i [\sum_{j=1}^{N} P_{ij} < w, x_{ij} > + b] \geq 1 - t_i \\
\Rightarrow & y_i [\sum_{j=1}^{N} P_{ij} \sum_{k=1}^{l} \alpha_k y_k \sum_{d=1}^{N} P_{kd} < x_{kd}, x_{ij} > + b] \geq 1 - t_i \\
\Rightarrow & y_i [\sum_{k=1}^{l} \alpha_k y_k < \sum_{j=1}^{N} P_{ij} x_{ij}, \sum_{d=1}^{N} P_{kd} x_{kd} > + b] \geq 1 - t_i \\
\Rightarrow & y_i y_k \sum_{k=1}^{l} \alpha_k < \sum_{j=1}^{N} P_{ij} x_{ij}, \sum_{d=1}^{N} P_{kd} x_{kd} > + y_i b \geq 1 - t_i \\
\Rightarrow & \tilde{K}_i \alpha + y_i b \geq 1 - t_i
\end{aligned}
\tag{5.8}
$$

where, $\tilde{k}_{ik} = y_i y_k < \sum_{j=1}^{N} P_{ij} x_{ij}, \sum_{d=1}^{N} P_{kd} x_{kd} >$.

From equations (5.7) and (5.8), the primal problem (5.3) becomes,

$$\begin{aligned} \min_{\alpha,b,t} \; & \frac{1}{2}\alpha^T \tilde{K}\alpha + C\sum_{i=1}^{l} t_i \\ s.t \quad & \tilde{K}_i\alpha + y_i b \geq 1 - t_i \\ & t_i \geq 0 \\ & i = 1,\Lambda\ ,l \end{aligned} \tag{5.9}$$

where, $\tilde{k}_{ik} = y_i y_k < \sum_{j=1}^{N} P_{ij} x_{ij}, \sum_{d=1}^{N} P_{kt} x_{kt} >$, and $\tilde{K}_i$ is the i-th row of the matrix $\tilde{K}$. The indices *i* and *k* are referring to the number of samples, and *j* and *d* to the number of replicated observations in the knowledge set, respectively. Problem (5.9) becomes a quadratic optimization problem.

5.2 Scenario Constrained Approach

In the previous section, we investigated the probability constrained approach using the kernelization method. Another issue that can be investigated is the selection of an appropriate probability distribution. The uncertainty is described through the knowledge set.

In this section we consider an extended version of the probability constrained approach. We call this extension a scenario constrained approach. Since the selection of a probability distribution is critical in the above problem, we adopt an idea of scenarios for different weights (coming from probability distribution function) for the

replicate measurements. Let q^S be the probability of a scenario *s* for selecting different weights (probability distributions) for the replicated measurements. Vector (5.6) can then be expressed as follows:

$$w = \sum_{i=1}^{l} \alpha_i y_i \sum_{s=1}^{S} q^S \sum_{j=1}^{N} P_{ij}{}^S x_{ij}{}^S \tag{5.10}$$

The quadratic part of the objective function in equation (5.3) *w* also can be replaced by the following:

$$\begin{aligned} \frac{1}{2} < w, w > &= \frac{1}{2} \sum_{i=1}^{l} \sum_{k=1}^{l} \alpha_i \alpha_k y_i y_k < \sum_{s=1}^{S} q^S \sum_{j=1}^{N} P_{ij}{}^S x_{ij}{}^S, \sum_{e=1}^{S} q^e \sum_{d=1}^{N} P_{kd}{}^e x_{kd}{}^e > \\ &= \frac{1}{2} \alpha^T \tilde{K} \alpha \end{aligned} \tag{5.11}$$

We define a modified kernel matrix as follows:

$$\tilde{k}_{ik} = y_i y_k < \sum_{s=1}^{S} q^S \sum_{j=1}^{N} P_{ij}{}^S x_{ij}{}^S, \sum_{e=1}^{S} q^e \sum_{d=1}^{N} P_{kd}{}^e x_{kd}{}^e > . \tag{5.12}$$

Note that the constraint of problem (5.8) becomes:

$$\begin{aligned} & y_i [\sum_{s=1}^{S} q^S \sum_{j=1}^{N} P_{ij}{}^S < w, x_{ij}{}^S > + b] \geq 1 - t_i \\ \Rightarrow & y_i [\sum_{s=1}^{S} q_a^S \sum_{j=1}^{N} P_{ij}{}^S \sum_{k=1}^{l} \alpha_k y_k \sum_{e=1}^{S} q^e \sum_{d=1}^{N} P_{kd}{}^e < x_{kd}{}^e, x_{ij}{}^e > + b] \geq 1 - t_i \\ \Rightarrow & y_i [\sum_{s=1}^{S} q^S \sum_{e=1}^{S} q^e \sum_{k=1}^{l} \alpha_k y_k < \sum_{j=1}^{N} P_{ij}{}^S x_{ij}{}^S, \sum_{d=1}^{N} P_{kd}{}^e x_{kd}{}^e > + b] \geq 1 - t_i \\ \Rightarrow & y_i y_k \sum_{k=1}^{l} \alpha_k \sum_{s=1}^{S} q^S \sum_{e=1}^{S} q^e < \sum_{j=1}^{N} P_{ij}{}^S x_{ij}{}^S, \sum_{d=1}^{N} P_{kd}{}^e x_{kd}{}^e > + y_i b \geq 1 - t_i \\ \Rightarrow & \tilde{K}_i \alpha + y_i b \geq 1 - t_i \end{aligned} \tag{5.13}$$

The sum of the scenario probabilities also must be 1, and each scenario probability is

not negative. The scenario *s* can be considered from the worst case to the best case, depending on prior knowledge. The final scenario constrained optimization problem we consider is as follows:

$$\begin{aligned} &\min_{\alpha,b,t} \frac{1}{2}\alpha^T \tilde{K}\alpha + C\sum_{i=1}^{l} t_i \\ &s.t \quad \tilde{K}_i\alpha + y_i b \geq 1 - t_i \\ &\qquad t_i \geq 0, \qquad i = 1,\Lambda\ ,l, \end{aligned} \tag{5.14}$$

where, $\tilde{k}_{ik} = y_i y_k < \sum_{s=1}^{S} q_a^S \sum_{j=1}^{N} P_{ij}{}^S x_{ij}{}^S , \sum_{e=1}^{S} q_b^e \sum_{d=1}^{N} P_{kd}{}^e x_{kd}{}^e >$.

5.3 Computational Experimentations

5.3.1 Computational Results for Probability Constrained Approach

Figure 5.2 illustrates a simple example of the probability constrained approach. By inserting the replicated data points into problem (5.9) as shown in Figure 5.2, we obtain:

$$\begin{aligned} \min_{\alpha_1,\alpha_2,b,t_1,t_2} \quad & \frac{1}{2}\alpha^T \tilde{K}\alpha + C(t_1+t_2) \\ s.t \quad & \tilde{K}_1\alpha + y_1 b \geq 1 - t_1 \\ & \tilde{K}_2\alpha + y_2 b \geq 1 - t_2 \\ & t_1, t_2 \geq 0 \qquad i = 1,2. \end{aligned} \tag{5.15}$$

Note that (5.15) is a quadratic optimization problem with five variables. The replicated points can be explained by the concept of perturbations of input data. This example has two knowledge sets representing two data points with three observations. To each observation we associate a weight (probability) within each knowledge set. Therefore, the matrix $\tilde{K}$ with $\tilde{k}_{ik} = y_i y_k < \sum_{j=1}^{N} P_{ij} x_{ij}, \sum_{d=1}^{N} P_{kt} x_{kt} >$ will be affected by the probability of the replicated points; that is, the center of gravity of the knowledge set will be shifted in terms of the weights of the observations.

Table 5. 1 Examples for four different probability sets.

	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{23}
Case 1	0.33	0.33	0.33	0.33	0.33	0.33
Case 2	0.1	0.2	0.7	0.5	0.2	0.3
Case 3	0.5	0.2	0.3	0.5	0.4	0.1
Case 4	0.4	0.3	0.3	0.2	0.8	0.0

To compare the behaviors of the separating hyperplanes, we consider four different weights in the observations (See Table 1). The probability p_{1i} is related to the class 1 (positive class), and p_{2i} refers to the class 2 (negative class), where $i = 1, \Lambda\ , 3$.

In Figure 5.2, for example p_{11} represents a weight of the first replicate measurement in class 1.

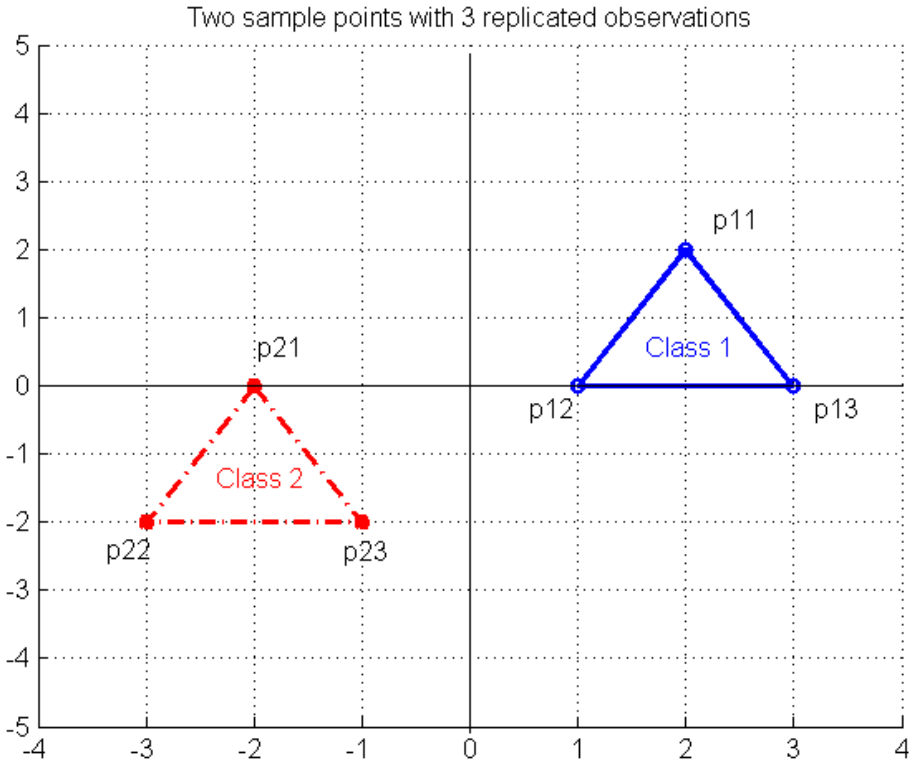

Figure 5. 2 Illustration of two sample points with three replicated observations.

Table 5.2 summarizes the solutions of the optimization problem (5.32) for the

four cases. We have obtained five parameters ($\alpha_1, \alpha_2, b, t_1$, and t_2) and the vector w has been calculated based on (5.6). From Table 5.2 and Figure 5.3, we can see that separating hyperplanes affects the different weights of observation. In case 4, we consider an extremely skewed case: the negative class data point is skewed on the left hand side from its center point (p_{23} =0). For this reason, the separating hyperplane has been shifted to the left, compared to the other cases. We can also see that the hyperplanes are rotated by the changes in data weights.

Based on the example of the probability constrained approach, we find the following important facts:

- The knowledge set will be given by the replicated observations.
- The center of gravity of the knowledge set shifts in terms of the weights (probabilities) of replicate measurements.
- The dimensionality of the problem is decreased, since the input samples are considered replicate measurements.

Table 5. 2 Solutions of probability constrained problem.

	α_1	α_2	b	t_1	t_2	w_1	w_2
Case 1	-0.0349	-0.0349	0	0	0	-0.1676	-0.0419
Case 2	-0.0274	-0.0274	0.0549	0	0	-0.1233	-0.0493
Case 3	-0.0216	-0.0216	1.0597	0	0	-0.1036	-0.0518
Case 4	-0.0442	-0.0442	0.3590	0	0	-0.1767	-0.0883

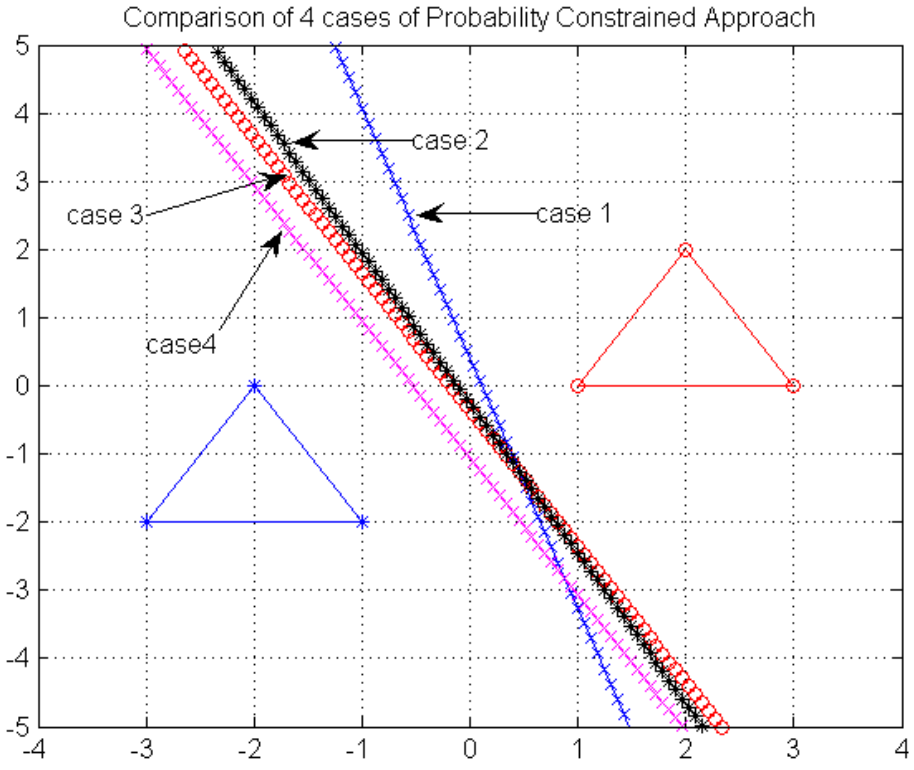

Figure 5. 3 Behavior of separating hyperplanes with four different cases.

To observe the behavior of the probability constrained problem in the real data, tornado data was considered. For the training, we selected 3 replicate data points (same day observations) from January to May. We obtained 15 tornado and 15 non-tornado data for the training, and used 952 testing data (50% of tornado and 50% of non-tornado).

Four cases of different weight in Table 5.1 were used for the experiments, and the results are shown in Table 5.3. Case 1 was considered with the same weight for every replicate measurement, and obtained same misclassification errors as traditional SVM solutions. Case 3 shows the worst misclassification error in comparison to the other cases. The interesting result is shown in case 2. The misclassification error in case 2 (16.4%) is decreased about five percent compared with traditional SVM solution (21.9%). In conclusion, if we select appropriate weights for the replicate measurement, reduced misclassification error can be obtained.

Table 5. 3 Misclassification errors with four different cases.

Cases	Case 1	Case 2	Case 3	Case 4	SVM
Misclassification Error	0.2195	0.1638	0.7447	0.3813	0.2195

5.3.2 Computational Results for Scenario Constrained Approach

To illustrate the scenario constrained approach, two sample points with three replicate measurements described in section 5.3.1 are used. In the scenario constrained approach, the replicate measurements were given with different scenarios in addition to their weights. For example, the first replicate measurement was collected by the first scenario, which had a higher probability in the positive class. In this example, three scenarios were considered:

- Scenario 1: more weights to the positive class (class 1)
- Scenario 2: equal weights to the two classes
- Scenario 3: more weights to the negative class (class 2).

The knowledge sets constructed by the replicate measurements were shifted by the given scenarios as shown in Figure 5.4. In this figure, the knowledge set of the positive class is shifted to the right side from the original set, while the negative knowledge set moved down to the left. Note that the centers of gravity of the knowledge sets were shifted as shown in Figure 5.4, and the separating hyperplane was

rotated to the left hand side while the margin was increased. In an application problem, the scenarios are given from the experts based on their prior knowledge.

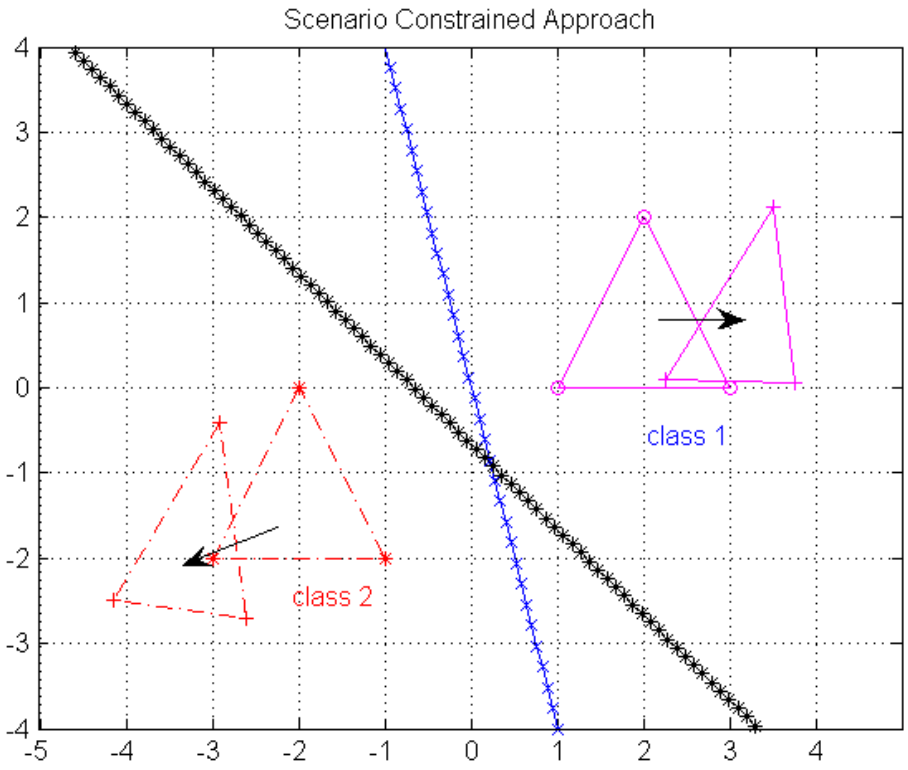

Figure 5. 4 Scenario constrained approach: knowledge set and the separating hyperplane have shifted by the scenarios.

CHAPTER 6. Conclusions and Future Research

6.1 Conclusions

In this research, we have developed a new robust optimization model for solving SVM learning problems, where we consider the perturbations of the parameters as well as the perturbations of input data. We illustrated examples for the three cases of possible perturbations, and showed how the SVM solution is influenced by the perturbation of input data, parameters, or both. In the classification and regression problem, the perturbation of input, in general, is more sensitive than the perturbation of parameters. However, if we consider the perturbation of input data and parameters, the effects make the solution less sensitive. In the case of perturbations of the input data with uniform spherical uncertainties, we observe that the resulting separating line is the SVM solution. This occurs when the uncertainty parameter η is smaller or equal to the margin of separation. This advocates the robustness of the SVM solution. Our model can be a basis for real-world problems that check the robustness

and feasibility of the SVM model under bounded perturbations.

The probability constrained approach and scenario constrained approach are investigated as alternatives to the robust optimization approach. The replicated measurements construct the knowledge set, which can be replaced with the uncertainty set in the robust optimization approach. This approach has an advantage when a small set of replicate measurements is given.

6.2 Future Research

There are several ways to measure uncertainties. Depending on the definition of the uncertainty (sphere, ellipsoid, rectangle, or a bounded convex set), a resulting classifier will be changed. Further research will handle more complicated uncertainties. Even though we consider several concepts of the uncertainty set and approaches, there still exist several assumptions - such as probability distributions and scenarios. If we consider the probability distribution and scenario as parameters, the problem will become more complicated.

In particular, when we do not have any information of prior knowledge for the data, we should consider the weights or probability distribution as parameters in our

models. Even if case 2 gives a good misclassification error in section 5.3.1, the optimal weight still is not guaranteed. The probability constrained approach suggested in 5.3.1 can be extended to the nonlinear optimization problem when the weight (probability) vector is considered as parameters. In future research, the weight (probability) for the replicate measurement will be added as an extra constraint in the previous probability constrained problem.

REFERENCES

[1] F. Alizadeh and D. Goldfarb. “Second-order cone programming.” *Mathematical Programming*, vol. 95, no. 1, pp. 3–51, 2003.

[2] C. Bhattacharyya. “Robust Classification of noisy data using Second Order Cone Programming approach.” *In Proceedings International Conference on Intelligent Sensing and Information Processing*, pp. 433-438, Chennai, India. 2004.

[3] C. Bhattacharyya, L.R. Grate, M.I. Jordan, L. EL. Ghaoui, and I.S. Mian, “Robust Sparse Hyperplane Classifiers: Application to Uncertain Molecular Profiling Data.” *Journal of Computational Biology*, vol. 11, no 6, pp. 1073-1089, 2004.

[4] M. Bazarra and C.M. Shetty. *Nonlinear programming.* New York, John Wiley, 1979.

[5] A. Ben-Tal and A. Nemirovski. “Robust convex optimization.” *Mathematics of Operations Research,* vol. 23, no 4, pp. 769-805, 1998.

[6] A. Ben-Tal and A. Nemirovski. “Robust solutions of uncertain linear programs.” *Operations Research Letters,* vol. 25, no 1, pp. 1-13, 1999.

[7] A. Ben-Tal and A. Nemirovski. “Robust solutions of Linear Programming problems contaminated with uncertain data.” *Mathematical Programming*, vol. 88, pp. 411-424, 2000.

[8] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1995.

[9] D. Bertsimas, D. Pachamanova and M. Sim. “Robust Linear Optimization under General Norms.” *Operations Research Letters*, vol. 32, no. 6, pp. 510-516, 2004

[10] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems.* New York: Springer-Verlag, 2000.

[11] O. Bousquet and A. Elisseff. “Stability and Generalization.” *Journal of Machine Learning Research,* vol. 2, pp. 399-526, 2002.

[12] S. Boyd and L. Vandenbrghe. *Convex Optimization.* Cambridge University Press, 2004.

[13] C. Burges. A Tutorial on Support Vector Machines for Pattern Classification. *Data mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.

[14] B. Boser, I. Guyon, and V. N. Vapnik. “A training algorithm for optimal margin classifiers.” *The Fifth Annual Workshop on Computational Learning Theory,* pp. 144-152, 1992.

[15] G. Calafiore and M. C. Campi. “Uncertain convex programs: randomized solutions and confidence levels.” *Mathematical Programming*, vol. 102, pp. 25-46, 2005.

[16] C. Cortes and V. Vapnik. “Support-vector networks.” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[17] J. Chinneck and K. Ramadan. “Linear programming with interval coefficients.” *Journal of the Operational Research Society*, vol. 51, pp. 209-220, 2000.

[18] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge: Cambridge University Press; 2000.

[19] A. Ding, X. Zhao, and L Jiao. ”Traffic Flow Time Series Prediction Based On Statistics Learning Theory.” *In processing of the IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 727-730, 2002.

[20] L. E. Ghaoui and H. Lebret. “Robust solutions to least-squares problems with uncertain data.” *SIAM J. on Matrix Analysis and Applications*, vol. 18, no 4, pp. 1035−1064, 1997.

[21] L. E. Ghaoui, G. Lanckriet and G. Natsoulis. “Robust Classification with Interval Data.” *In Technical Report CSD-03-1279 Division of Computer Science*, University of California, Berkeley, 2003.

[22] S. Haykin, *Neural Network: A Comprehensive Foundation.* Prentice Hall. New Jersey, pp. 331-332, 1999.

[23] K. Holmström. “The Tomlab Optimization Environment in Matlab .” *Advanced Modeling and Optimization*, vol 1, pp. 47-69, 1999.

[24] J.W.C. van Lint, S.P. Hoogendoorn and H. J. van Zuylen, “Robust and adaptive travel time prediction with neural networks,” *In proceedings of the 6th annual TRAIL Congress (part 2)*, December 2000.

[25] M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret. “Applications of second-order cone programming.” *Linear Algebra and Applications*, vol. 284, pp. 193-228, 1998.

[26] O. L. Mangasarian and W. H. Wolberg: "Cancer diagnosis via linear programming." *SIAM News*, vol. 23, no 5, pp 1 & 18, 1990.

[27] F. Oustry, L. El Ghaoui, and H. Lebret. “Robust Solutions to Uncertain Semidefinite Programs.” *SIAM J. on Optimization*, 1998.

[28] B. Schölkopf and A. J. Smola. *Learning with Kernels.* Cambridge, Massachusetts, The MIT Press, 2002.

[29] T. B. Trafalis and S. A. Alwazzi. "Robust Optimization in Support Vector Machine Training with Bounded Errors", *In proceedings of the International Joint Conference On Neural Networks*, IEEE Press, pp. 2039-2042, 2003a.

[30] T. B. Trafalis and S. A. Alwazzi. “Robust Support Vector Regression and Applications.” *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 13, pp. 181-186, 2003b.

[31] T. B. Trafalis and R.C. Gilbert. “Robust Classification and Regression Using Support Vector Machines.” *In processing of European Journal of Operational Research*, 2006.

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

[33] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[34] S. J. Wright. “Primal-dual Interior-Point Methods.” *Society for Industrial and Applied Mathematics (SIAM)*, Philadelphia, PA, 1997.

[35] C. H. Wu, J. M. Ho, and D. T. Lee, “Travel-time prediction with support vector regression.” *Intelligent Transportation Systems*, vol. 5, pp. 276-281, 2004.

APPENDICES

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% Chapter 4.1 SVC problems (AND Problem)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Perturbation of input data (AND problem)
function f=funD(x,eta1)
% Reassign the variables.
x1=x(1);
x2=x(2);
% Calculate the objective function.
f=0.5*sqrt(x1^2+x2^2);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [f,g]=nonlinD(x, eta1)
% Reassign the variables.
x1=x(1);
x2=x(2);
x3=x(3);
% Nonlinear inequalities
f(1)=-x1-x2-x3+eta1*sqrt(x1^2+x2^2)+1;
f(2)=x1-x2+x3+eta1*sqrt(x1^2+x2^2)+1;
f(3)=-x1+x2+x3+eta1*sqrt(x1^2+x2^2)+1;
f(4)=-x1-x2+x3+eta1*sqrt(x1^2+x2^2)+1;
% Nonlinear equalities
g=[];
end
% Find the solution
X=fmincon('funD',[0 0 0],[],[],[],[],[],[],'nonlinD',options,0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Perturbation of parameter (AND problem)
function f=funP(x,eta)
% Reassign the variables.
x1=x(1);
x2=x(2);

% Calculate the objective function.
f=0.5*sqrt(x1^2+x2^2);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [f,g]=nonlinP(x,eta)
% Reassign the variables.
x1=x(1);
x2=x(2);
x3=x(3);
% Nonlinear inequalities
f(1)=-2*x1-2*x2-x3+eta+1;
f(2)=2*x1-2*x2+x3+eta+1;
f(3)=-2*x1+2*x2+x3+eta+1;
f(4)=-2*x1-2*x2+x3+eta+1;

% Nonlinear equalities
g=[];
end

% Find the solution
X=fmincon('funP',[0 0 0],[],[],[],[],[],[],'nonlinP',options,0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% Chapter 4.1 SVC problems (XOR Problem)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Perturbation of input data (XOR problem)
function y=funD_XOR(x,eta)
% Reassign the variables.
x1=x(1);
x2=x(2);
x3=x(3);
x4=x(4);
x5=x(5);

% Caculate kernel matrix * Lagrange multipliers (Alpha)
m=81*(x1^2+x2^2+x3^2+x4^2)-2*(x1*x2+x1*x3+x2*x4+x3*x4)+98*(x1*x4+x2*x3);

% Calculate the objective function.
y= 0.5*m;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Perturbation of input data (XOR problem)
function [c,ce]=nonlinD_XOR(x,eta)
% Reassign the variables.
x1=x(1);
x2=x(2);
x3=x(3);
x4=x(4);
x5=x(5);

% Caculate kernel matrix * Lagrange multipliers (Alpha)
```

```
m=81*(x1^2+x2^2+x3^2+x4^2)-2*(x1*x2+x1*x3+x2*x4+x3*x4)+98*(x1*x4+x2*x3);


% Nonlinear inequalities
c(1)=(-1)*(81*x1-x2-x3+49*x4+x5)+eta*sqrt(m)+1;
c(2)=     (-x1+81*x2+49*x3-x4+x5)+eta*sqrt(m)+1;
c(3)=     (-x1+49*x2+81*x3-x4+x5)+eta*sqrt(m)+1;
c(4)=(-1)*(49*x1-x2-x3+81*x4+x5)+eta*sqrt(m)+1;

% Nonlinear equalities
ce=[];% No nonlinear equality constraints
end


% Find the solution
X=fmincon('funD_XOR',x0,[],[],[],[],[],[],' nonlinD_XOR',options,0.2,K)
% K is a kernel matrix.



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Chapter 4. Computational Results for Sensitivity and Robust Analysis Applied to
SVC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%  Case 1: Perturbation of Input Data %%%%%%%%%%%%%%%%%%%%%%%

clear all
data = load('breastcancerwisconsin.txt');
[v1 w]=size(data);
v=round(.5*v1); % 50% of training
x=data(1:v,1:w-1);
y=data(1:v,w);
tstx=data(v+1:v1,1:w-1);
```

```
tsty=data(v+1:v1,w);

ker='poly';
par=2;
eta=0.0;
[alpha,b]= chapter4_svc(x,y,ker,par,eta);

[ys,yt]=testsvc(x,tstx,ker,par,alpha,b);
error=find(yt'~=tsty);
error=length(error)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [alpha,b]=chapter4_svc(x,y,ker,par,C,eta);
[m n]= size(x);

eta=1.5
k = zeros(m,m);
ker='poly';
par=2;
k1=kernel(x', ker, par);
for i = 1:m
    for j = 1:m
        k(i,j)=y(i)*y(j)*k1(i,j);
    end
end
k;

x0=zeros(2*m+1,1);
lb2=zeros(m,1);
lb1=-inf*ones(m+1,1);
ubc=-ones(m,1);

if eta~=0
```

```
  %if 1
   Prob = conAssign(‘obj_1’,‘obj_2’ ,[],[],[lb1;lb2],[],'chapter4',x0,[],[],[],...
   [],[],’‘‘con_1’,’‘‘con_2’,[],[],[],[ubc]);
else
  Prob = conAssign(‘obj_1’,'obj_2’ ,[],[],[lb1;lb2],[],' chapter4',x0,[],[],[],...
         [],[],'con_11',' con_22',[],[],[],[ubc]);
end
Prob.user.H=k;
Prob.user.z=z;
Prob.user.C=C;
Prob.user.eta=eta;

Return=tomRun('snopt', Prob,[])
alpha= Return.x_k(1:m);
margin=1/sqrt(alpha'*k*alpha)
b= Return.x_k(m+1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ys,yt]=testsvc(x,tstx,ker,par,alpha,b)
    yt=[];
    m = size(tstx,1);
    k=kernel(tstx',x',ker,par);
    ys=(k*alpha)+b*ones(m,1);

for i=1:m
    if ys(i)<=0
        yt(i)=-1;
    else
        yt(i)=1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [f]=obj_1(x,Prob);
k=Prob.user.H;
z=Prob.user.z;
C=Prob.user.C;
m=size(x,1);
n = 0.5*(m - 1);
f= 0.5* x(1:n)'*k*x(1:n)+ C*sum(x(n+2:m);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gradient of objective function %%%%%%%%%%%%%%%%%%%

function [J]=obj_2(x,Prob);
k=Prob.user.H;
C=Prob.user.C;
m=length(x);
n = 0.5*(m-1);
J=zeros(length(x),1);
J(1:n) = x(1:n)'*k;
J(n+1)=0;
J(n+2:m)=C*ones(n,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [g]=con_1(x,Prob);
k=Prob.user.H;
z=Prob.user.z;
m=length(x);
n = 0.5*(m-1);
s = sqrt(x(1:n)'*k*x(1:n));
kx = k'*x(1:n);
slack=zeros(n,1);
for j=1:n
slack(j)=x(n+1+j);
end
g=s -kx- z.x(n+1)+slack;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [J]=con_2(x,Prob);
k=Prob.user.H;
z=Prob.user.z;
eta=Prob.user.eta;
m=length(x);
n = 0.5*(m-1);
J=zeros(n,length(x));
a = x(1:n)'*k*x(1:n);
x2k=a+1e-10*eye(size(a));
s  = eta*sqrt(x2k);
kx = x(1:n)'*k;
J(1:n,1:n) = -diag(z)*k';
J(1:n,n+1) = -z;
J(1:n,n+2:m)=-eye(n);
J(1:n,1:n) = J(1:n,1:n)-ones(n,1)*( eta/sqrt(x2k))*kx;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%% Perturbation of parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [g]=svmcon_par(x,Prob);
k=Prob.user.H;
z=Prob.user.z;
m=length(x);
n = 0.5*(m-1);
s = eta*Rad;

kx = k'*x(1:n);
slack=zeros(n,1);
for j=1:n
slack(j)=x(n+1+j);
end
```

```
g= s -kx- z.x(n+1)+slack;

%%%%%%%%%%%%%%%%%% Perturbation of input and  parameters %%%%%%%%%%%%%%%%%%%%%%%%%%

function [g]=svmcon_par(x,Prob);
k=Prob.user.H;
z=Prob.user.z;
m=length(x);
n = 0.5*(m-1);
s1 = sqrt(x(1:n)'*k*x(1:n));
s2 = eta*Rad;

kx = k'*x(1:n);
slack=zeros(n,1);
for j=1:n
slack(j)=x(n+1+j);
end
g= s1+s2 -kx- z.x(n+1)+slack;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%  Case 2: Perturbation of Parameters %%%%%%%%%%%%%%%%%%%%%%%%

function f = obj_124(x, Prob)

k=Prob.user.k;
y=Prob.user.y;
m=length(x)-1;
f= .5 * x(1:m)'*k*x(1:m);

%%%%%%%%%%%%%%%%%%%%%%%%%%    Constraint of the Problem %%%%%%%%%%%%%%%%%%%%%%%%%%%
function c = con_124(x, Prob)

k=Prob.user.k;
y=Prob.user.y;
```

```
eta=Prob.user.eta;
R=Prob.user.R;
m=length(x)-1;
c = -k*x(1:m) - y.*x(m+1) + eta*R*ones(m,1);


%%%%%%%%%%%%%  Case 3: Perturbation of Input Data and Parameters  %%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%    Constraint of the Problem %%%%%%%%%%%%%%%%%%%%%%%%%%%
function c = con_124(x, Prob)

k=Prob.user.k;
y=Prob.user.y;
eta1=Prob.user.eta1;
eta2=Prob.user.eta2;
R=Prob.user.R;
m=length(x)-1;

c = -k*x(1:m) - y.*x(m+1) +
eta1.*ones(m,1)*sqrt(x(1:m)'*k*x(1:m))+eta2*R.*ones(m,1);




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Chapter 4. Computational Results for Sensitivity and Robust Analysis Applied to
SVR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%% Case 1: Perturbation of Input Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
data=xlsread('2002a1.xls');
data=data(2:end,2:end);
```

```
p q]= size(data);
valueerror=[];
cumvalerror=[];
predict=[];
l=144;
tstY=[];
error=[];
for i=1:l %
        x=data(i,1:15)';
        y=data(i,16)*ones(size(x));
        tstX=data(i,17:27)';
        tstY=data(i,28)*ones(size(tstX));
ker='poly';
par=2;
C=100;
epsi=.1;
eta=5;  % eta and epsi are changad from 0 to 5
% e=1;
p=2;
[alpha,b]=traffic_reg(x,y,ker,par,epsi,eta);
 [yt] = out_svr(x,y,tstX,ker,p,alpha,b);
avg1=[mean(yt)];
predict=[predict avg1];
end

predict=predict';
save 2002a1.txt predict -ASCII


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% Compute parameters of alpha, b, and t %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [alpha,b]=traffic_reg(x,y,ker,par,epsi,eta)
[m,c] = size(x);

k = zeros(m,m);
k=kernel(x',ker,par);
z=y;

x0=zeros(2+m,1); % initial setup
lb2=zeros(1,1);
lb1=-inf*ones(m+1,1);
ubc1=epsi*ones(m,1);
ubc2=epsi*ones(m,1);
if eta~=0
    Prob = ...
conAssign('traffic_regof','traffic_regof_g',[],[],[lb1;lb2],[],'con2',x0,[],[],[],.
..
    [],[],'traffic_regcon','traffic_regcon_dc',[],[],[],[ubc1;ubc2]);
else
   Prob = ...
conAssign('traffic_regof','traffic_regof_g',[],[],[lb1;lb2],[],'con2',x0,[],[],[],.
..
           [],[],'traffic_regcon1','traffic_regcon1_dc',[],[],[],[ubc1;ubc2]);
end
Prob.user.H=k;
Prob.user.z=z;
Prob.user.eta=eta;

R=tomRun('snopt', Prob, []);
alpha=R.x_k(1:m);
b=R.x_k(m+1);
t=R.x_k(m+2);
clear R;
clear k;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%   Caluculate SVR Output   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 function [yt] = out_svr(x,Y,tstX,ker,p,alpha,b)
% This part is used by Steve Gunn (srg@ecs.soton.ac.uk)

    n = size(x,1);
    m = size(tstX,1);
    k=kernel(tstX',x',ker,p);
    yt = (k*alpha + b);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%% Objective Function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function f=traffic_regof(x,Prob)
m=size(x,1);
n=(m-2);
sum_slack=sum(x(n+2));
f=sum_slack;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%% First derivative of objective function %%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function g=traffic_regof_g(x,Prob)
m=length(x);
```

```
n = (m-2);
g=zeros(length(x),1);
g(1:n+1) =zeros(n+1,1);
g(n+2)=ones(1,1);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%% constraints function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


[g]=traffic_regcon(x,Prob);
m=length(x);
n=(m-2);
k=Prob.user.H;
y=Prob.user.z;
eta=Prob.user.eta;

x2k = x(1:n)'*k*x(1:n);
s = eta*sqrt(x2k);
kx = k'*x(1:n);
g=[y-kx+s-x(n+1)-x(n+2);-y+kx+s+x(n+1)-x(n+2)];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%    Case 2: Perturbation of Parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear all
data=xlsread('2002a1.xls');
data=data(2:end,2:end);
```

```
[p q]= size(data);

valueerror=[];
cumvalerror=[];
predict=[];
l=144;
tstY=[];
error=[];

for i=1:l
        x=data(i,1:15)';
        y=data(i,16)*ones(size(x));
        tstX=data(i,17:27)';
        tstY=data(i,28)*ones(size(tstX));
        ker='poly';
        par=2;

C=100;
epsi=.5;
eta=5;
Rad=3; % Radius considered 1,2,and 3
p=2;
[alpha,b]=traffic_reg(x,y,ker,par,epsi,eta,Rad);
[yt] = out_svr(x,y,tstX,ker,p,alpha,b);
avg1=[mean(yt)];
i;
predict=[predict avg1];
end
predict=predict';
save 2002a1_R3_5.txt predict -ASCII

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [alpha,b]=traffic_reg(x,y,ker,par,epsi,eta,Rad)
 [m,c] = size(x);
k = zeros(m,m);
k=kernel(x',ker,par);
z=y;

x0=zeros(2+m,1);
lb2=zeros(1,1);
lb1=-inf*ones(m+1,1);
ubc1=epsi*ones(m,1);
ubc2=epsi*ones(m,1);
if eta~=0
    Prob                                                                    =
conAssign('traffic_regof','traffic_regof_g',[],[],[lb1;lb2],[],'con2',x0,[],[],[],.
..
    [],[],'traffic_regcon','traffic_regcon_dc',[],[],[],[ubc1;ubc2]);
else
   Prob                                                                     =
conAssign('traffic_regof','traffic_regof_g',[],[],[lb1;lb2],[],'con2',x0,[],[],[],.
..
           [],[],'traffic_regcon1','traffic_regcon1_dc',[],[],[],[ubc1;ubc2]);
end
Prob.user.H=k;
Prob.user.z=z;
Prob.user.eta=eta;
Prob.user.Rad=Rad;

R=tomRun('snopt', Prob, []);
alpha=R.x_k(1:m);
%pos=find(alpha>=0);
%xsup=x(pos,:);
b=R.x_k(m+1);
t=R.x_k(m+2);
```

```
%ts=R.x_k(m+3)
clear R;
clear k;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Objective Function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f=traffic_regof(x,Prob)
m=size(x,1);
n=(m-2);
sum_slack=sum(x(n+2));
f=sum_slack;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%% Problem constraints %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [g]=traffic_regcon(x,Prob);
m=length(x);
n=(m-2);
%n=(1/3)*(m-1);
k=Prob.user.H;
y=Prob.user.z;
eta=Prob.user.eta;
Rad=Prob.user.Rad;
x2k = x(1:n)'*k*x(1:n);
s = eta*Rad;
kx = k'*x(1:n);
g=[y-kx+s-x(n+1)-x(n+2);...
   -y+kx+s+x(n+1)-x(n+2)];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%% Case 3: Perturbation of Input Data and Parameters %%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
data=xlsread('2002a1.xls');
data=data(2:end,2:end);
[p q]= size(data);

valueerror=[];
cumvalerror=[];
predict=[];
l=144;
tstY=[];
error=[];

for i=1:l % l=142
        x=data(i,1:15)';
        y=data(i,16)*ones(size(x));
        tstX=data(i,17:27)';
        tstY=data(i,28)*ones(size(tstX));
        ker='poly';
par=2;
C=100;
epsi=.5;
eta1=5;
eta2=5;
Rad=1;
p=2;
[alpha,b]=traffic_reg(x,y,ker,par,epsi,eta1,eta2,Rad);
[yt] = out_svr(x,y,tstX,ker,p,alpha,b);
```

```
avg1=[mean(yt)];
i;
predict=[predict avg1];
end
predict=predict';
save 2002a1_5_5.txt predict -ASCII


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [alpha,b]=traffic_reg(x,y,ker,par,epsi,eta1,eta2,Rad)
 [m,c] = size(x);
k = zeros(m,m);
k=kernel(x',ker,par);
z=y;

x0=zeros(2+m,1);
lb2=zeros(1,1);
lb1=-inf*ones(m+1,1);
ubc1=epsi*ones(m,1);
ubc2=epsi*ones(m,1);
if eta1~=0
    Prob =
conAssign('traffic_regof','traffic_regof_g',[],[],[lb1;lb2],[],'con2',x0,[],[],[],.
..
    [],[],'traffic_regcon','traffic_regcon_dc',[],[],[],[ubc1;ubc2]);
else
   Prob =
conAssign('traffic_regof','traffic_regof_g',[],[],[lb1;lb2],[],'con2',x0,[],[],[],.
..
           [],[],'traffic_regcon1','traffic_regcon1_dc',[],[],[],[ubc1;ubc2]);
end
Prob.user.H=k;
```

```
Prob.user.z=z;
Prob.user.eta1=eta1;
Prob.user.eta2=eta2;
Prob.user.Rad=Rad;

R=tomRun('snopt', Prob, []);
alpha=R.x_k(1:m);
b=R.x_k(m+1);
t=R.x_k(m+2);
clear R;
clear k;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%% Problem constraints %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [g]=traffic_regcon(x,Prob);
m=length(x);
n=(m-2);
%n=(1/3)*(m-1);
k=Prob.user.H;
y=Prob.user.z;
eta1=Prob.user.eta1;
eta2=Prob.user.eta2;
Rad=Prob.user.Rad;
x2k = x(1:n)'*k*x(1:n);
s = eta1*sqrt(x2k);
kx = k'*x(1:n);
g=[y-kx+s-x(n+1)-x(n+2)+eta2*Rad+eta1*eta2;...
    -y+kx+s+x(n+1)-x(n+2)+eta2*Rad+eta1*eta2];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%% Chapter 5.1 Probability Constrained Problem %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%% Case 1 %%%%%
clear all
   p1=[0.1 0.2 0.7]';
   p2=[0.5 0.4 0.1]';
    x11=[2 2; 1 0; 3 0];
    x22=[-2 0; -3 -2; -1 -2];
    X1=p1'*x11;
    X2=p2'*x22;
    X=[X1;X2];
    Y=[1 -1]';
   n = size(X,1);
   H = zeros(n,n);
    for i=1:n
       for j=1:n
          H1(i,j) = Y(i)*Y(j)*ker('poly',X(i,:),X(j,:));
       end
    end
col=zeros(2,3);
H=[H col];
row=zeros(3,5);
K=[H;row]
C=1000;
f=[0 0 0 C*1 C*1];
A=[ H1(1,:) -1 -1 0;
    H1(2,:) 1 0 -1;
    0 0 0 -1 0;
    0 0 0 0 -1];
b=[-1 -1 0 0];
x0=[0 0 0 0 0];
```

```
sol = quadprog(K,f,A,b,[],[],[],[],x0)
w=sol(1)*Y(1).*X1+sol(2)*Y(2).*X2

x1=[2 2; 1 0; 3 0; 2 2];
x2=[-2 0; -3 -2; -1 -2; -2 0];

 s=-w(1)/w(2);
 x=linspace(-3,3);
y=s*x-sol(3);
 plot(x,y,'x-') % Blue
 hold on
plot(x2(:,1),x2(:,2),'-.r*'); hold on;
plot(x1(:,1),x1(:,2),'-mo');
 grid on
axis([-4 4 -5 5])
hold off
% title ('Robust LP with Polyhedral Uncertainty (ex.1)');
title ('Comparison of 4 cases of Probability Constrained Approach');

hold on

%%%%% Case 2 %%%%%
 p1=[0.1 0.2 0.7]';
 p2=[0.2 0.4 0.4]';

%%%%% Case 3 %%%%%
p1=[0.5 0.2 0.3]';
p2=[0.5 0.4 0.1]';

%%%%% Case 4 %%%%%
p1=[0.4 0.3 0.3]';
p2=[0.2 0.8 0.0]';
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%% Tornado data analysis %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
tst=xlsread('tst5.xls');
trn=xlsread('E:₩data_tor.xls');
    p1=[0.2 0.3 0.5]';
    p2=[0.4 0.3 0.3]';
    p1=repmat(p1,[5,1]);
    p2=repmat(p2,[5,1]);
    p1=repmat(p1,[1,25]);
    p2=repmat(p2,[1,25]);
    X1=p1.*trn(1:15,1:end-1);
    X2=p2.*trn(16:30,1:end-1);
    trnX=[X1;X2];

[k l]=size(trn);
trnY=trn(:,end);

tstX=tst(:,5:29);
tstY=tst(:,end);

[trnX] = prestd(trnX')';
[tstX] = prestd(tstX')';

[nsv, alpha, bias] = svc(trnX,trnY,'poly',100);
err = svcerror(trnX,trnY,tstX,tstY,'poly',alpha,bias);
out = svcoutput(trnX,tstX,tstX,'poly',alpha,bias);
 [sign(out) tstY]
num_correct=size(find(sign(out)== tstY))
size_of_test=size(out)
```