Journal of Machine Learning Research 7 (2006) 1283–1314 Submitted 7/05; Published 7/06

Second Order Cone Programming Approaches for Handling Missing and Uncertain Data

Pannagadatta K. Shivaswamy PANNAGA@CS.COLUMBIA.EDU
Computer Science
Columbia University
New York, 10027, USA

Chiranjib Bhattacharyya CHIRU@CSA.IISC.ERNET.IN
Department of Computer Science and Automation
Indian Institute of Science
Bangalore, 560 012, India

Alexander J. Smola ALEX.SMOLA@NICTA.COM.AU
Statistical Machine Learning Program
National ICT Australia and ANU
Canberra, ACT 0200, Australia

Editors: Kristin P. Bennett and Emilio Parrado-Hernández

Abstract

We propose a novel second order cone programming formulation for designing robust classifiers which can handle uncertainty in observations. Similar formulations are also derived for designing regression functions which are robust to uncertainties in the regression setting. The proposed formulations are independent of the underlying distribution, requiring only the existence of second order moments. These formulations are then specialized to the case of missing values in observations for both classification and regression problems. Experiments show that the proposed formulations outperform imputation.

1. Introduction

Denote by $(x,y) \in \mathcal{X} \times \mathcal{Y}$ patterns with corresponding labels. The typical machine learning formulation only deals with the case where (x,y) are given *exactly*. Quite often, however, this is not the case — for instance in the case of missing values we may be able (using a secondary estimation procedure) to estimate the values of the missing variables, albeit with a certain degree of uncertainty. In other cases, the observations maybe systematically censored. In yet other cases the data may represent an entire equivalence class of observations (e.g. in optical character recognition all digits, their translates, small rotations, slanted versions, etc. bear the same label). It is therefore only natural to take the potential range of such data into account and design estimators accordingly. What we propose in the present paper goes beyond the traditional imputation strategy in the context of missing variables. Instead, we integrate the fact that some observations are not completely determined into the optimization problem itself, leading to convex programming formulations.

In the context of this paper we will assume that the uncertainty is only in the patterns x, e.g. some of its components maybe missing, and the labels y are known precisely whenever given. We first consider the problem of binary classification where the labels y can take two values, $\mathcal{Y} =$

$\{1,-1\}$. This problem was partially addressed in (Bhattacharyya et al., 2004b), where a second order cone programming (SOCP) formulation was derived to design a robust linear classifier when the uncertainty was described by multivariate normal distributions. Another related approach is the Total Support Vector Classification (TSVC) of Bi and Zhang (2004) who, starting from a very similar premise, end up with a non-convex problem with corresponding iterative procedure.

One of the main contributions of this paper is to generalize the results of Bhattacharyya et al. (2004b) by proposing a SOCP formulation for designing robust binary classifiers for arbitrary distributions having finite mean and covariance. This generalization is acheived by using a multivariate Chebychev inequality (Marshall and Olkin, 1960). We also show that the formulation achieves robustness by requiring that for every uncertain datapoint an ellipsoid should lie in the correct half-space. This geometric view immediately motivates various error measures which can serve as performance metrics. We also extend this approach to the multicategory case. Next we consider the problem of regression with uncertainty in the patterns x. Using Chebyshev inequalities two SOCP fromulations are derived, namely *Close to Mean* formulation and *Small Residual* formulation, which give linear regression functions robust to the uncertainty in x. This is another important contribution of this paper. As in the classification case the formulations can be interpreted geometrically suggesting various error measures. The proposed formulations are then applied to the problem of patterns having missing values both in the case of classification and regression. Experiments conducted on real world data sets show that the proposed formulations outperform imputations. We also propose a way to extend the proposed formulations to arbitrary feature spaces by using kernels for both classification and regression problems.

Outline: The paper is organised as follows: Section 2 introduces the problem of classification with uncertain data. In section 3 we make use of Chebyshev inequalities for multivariate random variable to obtain an SOCP which is one of the main contribution of the paper. We also show that same formulation could be obtained by assuming that the underlying uncertainty can be modeled by an ellipsoid. This geometrical insight is exploited for designing various error measures. A similar formulation is obtained for a normal distribution. Instead of an ellipsoid one can think of more general sets to describe uncertainty. One can tackle such formulations by constraint sampling methods. These constraint sampling methods along with other extensions are discussed in section 4. The other major contribution is discussed in section 5. Again using Chebyshev inequalities two different formulations are derived for regression in section 5 for handling uncertainty in x. As before the formulations motivate various error measures which are useful for comparison. In section 6 we specialize the formulations to the missing value problem both in the case of classification and regression. In section 7 nonlinear prediction functions are discussed. To compare the performance of the formulations numerical experiments were performed on various real world datasets. The results are compared favourably with the imputation based strategy, details are given in section 8. Finally we conclude in section 9.

2. Linear Classification by Hyperplanes

Assume that we have n observations (x_i, y_i) drawn iid (independently and identically distributed) from a distribution over $\mathcal{X} \times \mathcal{Y}$, where x_i is the i^{th} pattern and y_i is the corresponding label. In the following we will briefly review the SVM formulation when the observations are known with certainty and then consider the problem of uncertain observations.

2.1 Classification with Certainty

For simplicity assume that $\mathcal{Y} = \{\pm 1\}$ and $\mathcal{X} = \mathbb{R}^m$ with a finite m. For linearly separable datasets we can find a hyperplane $\langle w, x\rangle + b = 0$ [1] which separates the two classes and the corresponding classification rule is given by

$$f(x) = \operatorname{sgn}(\langle w, x\rangle + b).$$

One can compute the parameters of the hyperplane (w, b) by solving a quadratic optimization problem (see Cortes and Vapnik (1995))

$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 \tag{1a}$$

$$\text{subject to } y_i(\langle w, x_i\rangle + b) \geq 1 \qquad \text{for all } 1 \leq i \leq n, \tag{1b}$$

where $\|w\|$ is the euclidean norm.[2] In many cases, such separation is impossible. In this sense the constraints (1b) are hard. One can still construct a hyperplane by relaxing the constraints in (1). This leads to the following soft margin formulation with L_1 regularization (Bennett and Mangasarian, 1993; Cortes and Vapnik, 1995):

$$\underset{w,b,\xi}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{2a}$$

$$\text{subject to } y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \qquad \text{for all } 1 \leq i \leq n \tag{2b}$$

$$\xi_i \geq 0 \qquad \text{for all } 1 \leq i \leq n. \tag{2c}$$

The above formulation minimizes an upper bound on the number of errors. Errors occur when $\xi_i \geq 1$. The quantity $C\xi_i$ is the "penalty" for any data point x_i that either lies within the margin on the correct side of the hyperplane ($\xi_i \leq 1$) or on the wrong side of the hyperplane ($\xi_i > 1$).

One can re-formulate (2) as an SOCP by replacing the $\|w\|^2$ term in the objective (2a) by a constraint which upper bounds $\|w\|$ by a constant W. This yields

$$\underset{w,b,\xi}{\text{minimize}} \ \sum_{i=1}^{n}\xi_i \tag{3a}$$

$$\text{subject to } y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \qquad \text{for all } 1 \leq i \leq n \tag{3b}$$

$$\xi_i \geq 0 \qquad \text{for all } 1 \leq i \leq n \tag{3c}$$

$$\|w\| \leq W. \tag{3d}$$

Instead of C the formulation (3) uses a direct bound on $\|w\|$, namely W. One can show that for suitably chosen C and W the formulations (2) and (3) give the same optimal values of (w, b, ξ). Note that (3d) is a second order cone constraint (Lobo et al., 1998).[3] With this reformulation in mind we will, in the rest of the paper, deal with (2) and, with slight abuse of nomenclature, discuss SOCPs where the transformation from (2) to (3) is implicit.

1. $\langle a, b\rangle$ denotes the dot product between $a, b \in \mathcal{X}$. For $\mathcal{X} = \mathbb{R}^m$, $\langle a, b\rangle = a^\top b$. The formulations discussed in the paper holds for arbitrary Hilbert spaces with a suitably defined dot product $\langle ., .\rangle$.
2. The Euclidean norm for element $x \in \mathcal{X}$ is defined as $\|x\| = \sqrt{\langle x, x\rangle}$ where $\mathcal{X}$ is a Hilbert space.
3. Second order cones are given by inequalities in w which take the form $\|\Sigma w + c\| \leq \langle w, x\rangle + b$. In this case $c = 0$ and the cone contains a ray in the direction of $-w$, b determines the offset from the origin, and Σ determines the shape of the cone.

2.2 Classification Under Uncertainty

So far we assumed that the (x_i, y_i) pairs are known with certainty. In many situations this may not be the case. Suppose that instead of the pattern (x_i, y_i) we only have a distribution over x_i, that is x_i is a random variable. In this case we may replace (2b) by a probabilistic constraint

$$\Pr_{x_i}\{y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i\} \geq 1 - \kappa_i \text{ for all } 1 \leq i \leq n. \tag{4}$$

In other words, we require that the random variable x_i lies on the correct side of the hyperplane with probability greater than κ_i. For high values of κ_i, which is a user defined parameter in $(0,1]$, one can obtain a good classifier with a low probability of making errors.

Unless we make some further assumptions or approximations on (4) it will be rather difficult to solve it directly. For this purpose the following sections describe various approaches on how to deal with the optimization. We begin with the assumption that the second moments of x_i exist. In this case we may make use of Chebyshev inequalities (Marshall and Olkin, 1960) to obtain a SOCP.

2.3 Inequalities on Moments

The key tool are the following inequalities, which allow us to bound probabilities of misclassification subject to second order moment constraints on x. Markov's inequality states that if ξ is a random variable, $h : \mathbb{R} \to [0,\infty)$ and a is some positive constant then

$$\Pr\{h(\xi) \geq a\} \leq \frac{\mathbf{E}[h(\xi)]}{a}.$$

Consider the function $h(x) = x^2$. This yields

$$\Pr\{|\xi| \geq a\} \leq \frac{\mathbf{E}\left[\xi^2\right]}{a^2}. \tag{5}$$

Moreover, considering $h(x) = (x - \mathbf{E}[x])^2$ yields the Chebyshev inequality

$$\Pr\{|\xi - \mathbf{E}(\xi)| \geq a\} \leq \frac{\mathbf{Var}[\xi]}{a^2}. \tag{6}$$

Denote by $\bar{x}, \Sigma$ mean and variance of a random variable x. In this case the multivariate Chebyshev inequality (Marshall and Olkin, 1960; Lanckriet et al., 2002; Boyd and Vandenberghe, 2004) is given by

$$\sup_{x\sim(\bar{x},\Sigma)} \Pr\{\langle w, x\rangle \leq t\} = (1+d^2)^{-1} \text{ where } d^2 = \inf_{x|\langle x,w\rangle \leq t} (x-\bar{x})^\top \Sigma^{-1}(x-\bar{x}). \tag{7}$$

This bound always holds for a family of distributions having the same second order moments and in the worst case equality is attained. We will refer to the distribution corresponding to the worst case as the *worst distribution*. These bounds will be used to turn the linear inequalities used in Support Vector Machine classification and regression into inequalities which take the uncertainty of the observed random variables into account.

3. Classification

The main results of our work for the classification problem are presented in this section. Second order cone programming solutions are developed which can handle uncertainty in the observations.

3.1 Main Result

In order to make progress we need to specify properties of (4). Several settings come to mind and we will show that all of them lead to an SOCP.

Robust Formulation Assume that for each x_i we only know its mean $\bar{x_i}$ and variance Σ_i. In this case we want to be able to classify correctly even for the *worst distribution* in this class. Denote by $x \sim (\mu, \Sigma)$ a family of distributions which have a common mean and covariance, given by μ and Σ respectively. In this case (4) becomes

$$\inf_{x_i \sim (\bar{x_i}, \Sigma_i)} \Pr_{x_i} \left(y_i \left(\langle x_i, w \rangle + b \right) \geq 1 - \xi_i \right) \geq 1 - \kappa_i. \tag{8}$$

This means that even for the worst distribution we still classify x_i correctly with high probability $1 - \kappa_i$.

Normal Distribution Equally well, we might assume that x_i is, indeed, distributed according to a normal distribution with mean $\bar{x_i}$ and variance Σ_i. This should allow us to provide tighter bounds, as we have perfect knowledge on how x_i is distributed. In other words, we would like to solve the classification problem, where (4) becomes

$$\Pr_{x_i \sim \mathcal{N}(\bar{x_i}, \Sigma_i)} \left(y_i \left(\langle x_i, w \rangle + b \right) \geq 1 - \xi_i \right) \geq 1 - \kappa_i. \tag{9}$$

Using a Gaussian assumption on the underlying data allows one to use readily available techniques like EM (Dempster et al., 1977; Schneider, 2001) to impute the missing values.

It turns out that both (8) and (9) lead to the same optimization problem.

Theorem 1 *The classification problem with uncertainty, as described in (4) leads to the following second order cone program, when using constraints (8), (9):*

$$\underset{w,b,\xi}{\text{minimize}} \ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \tag{10a}$$

$$\text{subject to } y_i \left(\langle w, \bar{x_i} \rangle + b \right) \geq 1 - \xi_i + \gamma_i \left\| \Sigma_i^{\frac{1}{2}} w \right\| \qquad \text{for all } 1 \leq i \leq n \tag{10b}$$

$$\xi_i \geq 0 \qquad \text{for all } 1 \leq i \leq n, \tag{10c}$$

where $\Sigma^{\frac{1}{2}}$ is a symmetric square matrix and is the matrix square root of $\Sigma = \Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}}$.

More specifically, the following formula for γ_i hold:

- *In the robust case $\bar{x_i}$, Σ_i correspond to the presumed means and variances and*

$$\gamma_i = \sqrt{\kappa_i / (1 - \kappa_i)}. \tag{11}$$

- *In the normal distribution case, again $\bar{x_i}, \Sigma_i$ correspond to mean and variance. Moreover γ_i is given by the functional inverse of the normal CDF, that is*

$$\gamma_i = \phi^{-1}(\kappa_i) \text{ where } \phi(u) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} e^{-\frac{s^2}{2}} ds. \tag{12}$$

Note that for $\kappa_i < 0.5$ the functional inverse of the Gaussian cumulative distribution function becomes negative. This means that in those cases the joint optimization problem is *nonconvex*, as the second order cone constraint enters as a *concave* function. This is the problem that Bi and Zhang (2004) study. They find an iterative procedure which will converge to a local optimum. On the other hand, whenever $\gamma_i \geq 0$ we have a *convex* problem with unique minimum value.

As expected $\phi^{-1}(\kappa_i) < \sqrt{\frac{\kappa_i}{1-\kappa_i}}$. What this means in terms of our formulation is that, by making Gaussian assumption we only scale down the size of the uncertainty ellipsoid with respect to the Chebyshev bound.

Formulation (10) can be solved efficiently using various interior point optimization methods (Boyd and Vandenberghe, 2004; Lobo et al., 1998; Nesterov and Nemirovskii, 1993) with freely available solvers, such as SeDuMi (Sturm, 1999) making them attractive for large scale missing value problems.

3.2 Proof of Theorem 1

Robust Classification We can restate (8) as

$$\sup_{x\sim(\bar{x}_i,\Sigma_i)} \Pr_x \{y_i(\langle w,x\rangle + b) \leq 1-\xi_i\} \leq \kappa_i.$$

See that it is exactly equivalent to (8) and using Eq. (7) we can write

$$\sup_{x\sim(\bar{x}_i,\Sigma_i)} \Pr_x \{y_i(\langle w,x\rangle + b) \geq 1-\xi_i\} = (1+d^2)^{-1} \leq \kappa_i, \tag{13a}$$

$$\text{where, } d^2 = \inf_{x|y_i(\langle x,w\rangle+b)\leq 1-\xi_i} (x-\bar{x}_i)^\top \Sigma_i^{-1}(x-\bar{x}_i). \tag{13b}$$

Now we solve (13b) explicitly. In case $\bar{x}_i$ satisfies $y_i(\langle w,x_i\rangle + b) \geq 1-\xi_i$ then clearly the infimum in (13b) is zero. If not, d^2 is just the distance of the mean $\bar{x}_i$ from the hyperplane $y_i(\langle w,x_i\rangle + b) = 1-\xi_i$, that is

$$d^2 = \frac{y_i(\langle w,\bar{x}_i\rangle + b - 1 + \xi_i)}{\sqrt{w^\top \Sigma_i w}}. \tag{14}$$

The expression for d^2 in (14) when plugged into the requirement $\frac{1}{1+d^2} \leq \kappa_i$ gives (10b) where γ_i is given as in (11) thus proving the first part.

Normal Distribution Since projections of a normal distributions are themselves normal we may rewrite (9) as a scalar probabilistic constraint. We have

$$\Pr\left\{\frac{z_i-\bar{z}_i}{\sigma_{z_i}} \geq \frac{y_i b + \xi_i - 1 - \bar{z}_i}{\sigma_{z_i}}\right\} \quad \leq \quad \kappa_i, \tag{15}$$

where $z_i := -y_i\langle w,x_i\rangle$ is a normal random variable with mean $\bar{z}_i$ and variance $\sigma_{z_i}^2 := w^\top \Sigma_i w$. Consequently $(z_i-\bar{z}_i)/\sigma_{z_i}$ is a random variable with zero mean and unit variance and we can compute the lhs of (15) by evaluating the cumulative distribution function $\phi(x)$ for normal distributions. This makes (15) equivalent to the condition

$$\phi\left(\sigma_{z_i}^{-1}(y_i b + \xi_i - 1 - \bar{z}_i)\right) \geq \kappa_i,$$

which can be solved for the argument of ϕ.

3.3 Geometric Interpretation and Error Measures

The constraint (10b) can also be derived from a geometric viewpoint. Assume that x takes values in an ellipsoid with center $\bar{x}$, metric Σ and radius[4] γ, that is

$$x \in \mathcal{E}(\bar{x}, \Sigma, \gamma) := \left\{ x | \, (x-\bar{x})^\top \Sigma^{-1} (x-\bar{x}) \leq \gamma^2 \right\}. \tag{16}$$

The robustness criteria can be enforced by requiring that that we classify x correctly for all $x \in \mathcal{E}(\bar{x}, \Sigma, \gamma)$, that is

$$y(\langle x, w \rangle + b) \geq 1 - \xi \text{ for all } x \in \mathcal{E}(\bar{x}, \Sigma, \gamma). \tag{17}$$

In the subsequent section we will study other constraints than ellipsoid sets for x.

Lemma 2 *The optimization problem*

$$\underset{x}{\text{minimize}} \ \langle w, x \rangle \ \text{subject to } x \in \mathcal{E}(\bar{x}, \Sigma, \gamma)$$

has its minimum at $\bar{x} - \gamma \left(w^\top \Sigma w \right)^{-\frac{1}{2}} \Sigma w$ *with minimum value* $\langle \bar{x}, w \rangle - \gamma \left(w^\top \Sigma w \right)^{\frac{1}{2}}$. *Moreover, the maximum of* $(\langle w, x \rangle - \langle w, \bar{x} \rangle)$ *subject to* $x \in \mathcal{E}(\bar{x}, \Sigma, \gamma)$ *is given by* $\gamma \left\| \Sigma^{\frac{1}{2}} w \right\|$.

Proof We begin with the second optimization problem. Substituting $v := \Sigma^{-\frac{1}{2}} (x - \bar{x})$ one can see that the problem is equivalent to maximizing $\langle w, \Sigma^{\frac{1}{2}} v \rangle$ subject to $\|v\| \leq \gamma$. The latter is maximized for $v = \gamma \Sigma^{\frac{1}{2}} w / \left\| \Sigma^{\frac{1}{2}} w \right\|$ with maximum value $\gamma \left\| \Sigma^{\frac{1}{2}} w \right\|$. This proves the second claim.

The first claim follows from the observation that maximum and minimum of the second objective function match (up to a sign) and from the fact that the first objective function can be obtained form the second by a constant offset $\langle w, \bar{x} \rangle$. ■

This means that for fixed w the minimum of the lhs of (17) is given by

$$y_i (\langle \bar{x_i}, w \rangle + b) - \gamma_i \sqrt{w^\top \Sigma_i w}. \tag{18}$$

The parameter γ is a function of κ, and is given by (11) in the general case. For the normal case it is given by (12). We will now use this ellipsoidal view to derive quantities which can serve as performance measures on a test set.

Worst Case Error: given an uncertainty ellipsoid, we can have the following scenarios:

1. The centroid is classified correctly and the hyperplane does not cut the ellipsoid: The error is zero as all the points within the ellipsoid are classified correctly.

2. The centroid is misclassified and the hyperplane does not cut the ellipsoid: Here the error is 1 as all the points within the ellipsoid are misclassified.

3. The hyperplane cuts the ellipsoid. Here the worst case error is one as we can always find points within the uncertainty ellipsoid that get misclassified.

4. Note that we could as well dispose of γ by transforming $\Sigma \leftarrow \gamma^{-2} \Sigma$. The latter, however, leads to somewhat inconvenient notation.

Figure 1 illustrates these cases. It shows a scenario in which there is uncertainty in two of the features. Figure corresponds to those two dimensions. It shows three ellipsoids corresponding to the possible scenarios.

To decide whether the ellipsoid, $\mathcal{E}(\mu,\Sigma,\gamma)$, intersects the hyperplane, $w^\top x+b=0$, one needs to compute

$$z=\frac{w^\top\mu+b}{\sqrt{w^\top\Sigma w}}.$$

If $|z|\leq\gamma$ then the hyperplane intersects the ellipsoid, see (Bhattacharyya et al., 2004a). For an uncertain observation, i.e. given an ellipsoid, with the label y, the worst case error is given by

$$e_{wc}(\mathcal{E})=\begin{cases}1 & \text{if } yz<\gamma\\ 0 & \text{otherwise.}\end{cases}$$

Expected Error The previous measure is a pessimistic one. A more optimistic measure could be the expected error. We find out the volume of the ellipsoid on the wrong side of the hyperplane and use the ratio of this volume to the entire volume of the ellipsoid as the expected error measure. When the hyperplane doesn't cut the ellipsoid, expected error is either zero or one depending on whether the ellipsoid lies entirely on the correct side or entirely on the wrong side of the hyperplane. In some sense, this measure gives the expected error for each sample when there is uncertainty. In figure 1 we essentially take the fraction of the area of the shaded portion of the ellipsoid as the expected error measure. In all our experiments, this was done by generating large number of uniformly distributed points in the ellipsoid and then taking the fraction of the number of points on the correct side of the hyperplane to the total number of points generated.

4. Extensions

We now proceed to extending the optimization problem to a larger class of constraints. The following three modifications come to mind: (a) extension to multiclass classification, (b) extension of the setting to different types of set constraints, and (c) the use of constraint sampling to deal with nontrivial constraint sets

4.1 Multiclass Classification

An obvious and necessary extension of above optimization problems is to deal with multiclass classification. Given $y\in\mathcal{Y}$ one solves the an optimization problem maximizing the multiclass margin (Collins, 2002; Rätsch et al., 2002; Taskar et al., 2003):

$$\underset{w,\xi}{\text{minimize}}\ \sum_{i=1}^{n}\xi_i \tag{19a}$$

$$\text{subject to } \langle w_{y_i},x_i\rangle-\max_{y\neq y_i}\langle w_y,x_i\rangle\geq 1-\xi_i \text{ and } \xi_i\geq 0 \qquad \text{for all } 1\leq i\leq n \tag{19b}$$

$$\sum_{i=1}^{|\mathcal{Y}|}\|w_{y_i}\|^2\leq W^2. \tag{19c}$$

Here w_i are the weight vectors corresponding to each class. Taking square roots of (19c) yields a proper SOCP constraint on $w\in\mathbb{R}^{d\times|\mathcal{Y}|}$. Note that instead of (19b) we could also state $|\mathcal{Y}|-1$ linear

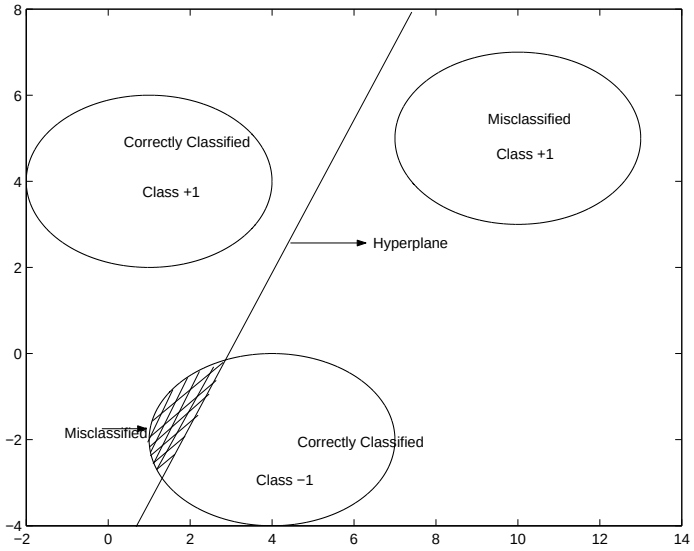

Figure 1: Three scenarios occurring when classifying a point: One of the unshaded ellipsoids lies entirely on the "correct" side of the hyperplane, the other lies entirely on the "wrong" side of the hyperplane. The third, partially shaded ellipsoid has parts on either sides. In the worst case we count the error for this pattern as one whereas in the expected case we count the error as the fraction of the volume (in this case area) on the "wrong" side as the error

inequalities on w_i according to each (y_i, y) combination. The latter allows us apply a reasoning analogous to that of Theorem 1 (we skip the proof as it is identical to that of Section 3.2 with small modifications for a union bound argument). This yields:

$$\underset{w,b,\xi}{\text{minimize}} \ \frac{1}{2}\sum_{i=1}^{|\mathcal{Y}|}\|w_i\|^2 + C\sum_{i=1}^{n}\xi_i \tag{20a}$$

$$\text{subject to } (\langle w_{y_i} - w_y, \overline{x_i}\rangle) \geq 1 - \xi_i + \gamma_i \left\| \Sigma_i^{\frac{1}{2}}(w_{y_i} - w_y) \right\| \qquad \text{for } 1 \leq i \leq n, y \neq y_i \tag{20b}$$

$$\xi_i \geq 0 \qquad \text{for } 1 \leq i \leq n. \tag{20c}$$

The key difference between (10) and (20) is that we have a set of $|\mathcal{Y}| - 1$ second order cone constraints per observation.

4.2 Set Constraints

The formulations presented so far can be broadly understood in the context of robust convex optimization (see Ben-Tal and Nemirovski (1998, 2001)). In the following we discuss a few related formulations which were proposed in the context of pattern classification. This subsection lists types of the constraint set and the kind of optimization problems used for solving SVM for the underlying constraint sets.

Note that we may rewrite the constraints on the classification as follows:

$$y_i(\langle x, w\rangle + b) \geq 1 - \xi_i \text{ for all } x \in S_i. \tag{21}$$

Here the sets S_i are given by $S_i = \mathcal{E}(\bar{x_i}, \Sigma_i, \gamma_i)$. This puts our optimization setting into the same category as the knowledge-based SVM (Fung et al., 2002) and SDP for invariances (Graepel and Herbrich, 2004), as all three deal with the above type of constraint (21), but the set S_i is different. More to the point, in (Graepel and Herbrich, 2004) $S_i = S(b_i, \beta)$ is a polynomial in β which describes the set of invariance transforms of x_i (such as distortion or translation). (Fung et al., 2002) define S_i to be a polyhedral "knowledge" set, specified by the intersection of linear constraints.

By the linearity of (21) it follows that if (21) holds for S_i then it also holds for coS_i, the convex hull of S_i. Such considerations suggest yet another optimization setting: instead of specifying a polyhedral set S_i by constraints we can also specify it by its vertices. Depending on S_i such a formulation may be computationally more efficient.

In particular if S_i is the convex hull of a set of generators x_{ij} as in

$$S_i = \text{co}\{x_{ij} \text{ for } 1 \leq j \leq m_i\}.$$

We can replace (21) by

$$y_i(\langle w, x_{ij}\rangle + b) \geq 1 - \xi_i \text{ for all } 1 \leq j \leq m_i.$$

In other words, enforcing constraints for the convex hull is equivalent to enforcing them for the *vertices* of the set. Note that the index ranges over j rather than i. Such a setting is useful e.g. in the case of range constraints, where variables are just given by interval boundaries. Table 1 summarizes the five cases. Clearly all the above constraints can be mixed and matched. More central is the notion of stating the problems via (21) as a starting point.

Table 1: Constraint sets and corresponding optimization problems.

Name	Set S_i	Optimization Problem
Plain SVM	$\{x_i\}$	Quadratic Program
Knowledge Based SVM	Polyhedral set	Quadratic Program
Invariances	trajectory of polynomial	Semidefinite Program
Normal Distribution	$\mathcal{E}(x_i, \Sigma_i, \gamma_i)$	Second Order Cone Program
Convex Hull	$\text{co}\{x_{ij} \; \forall \; 1 \leq j \leq m_i\}$	Quadratic Program

4.3 Constraint Sampling Approaches

In the cases of Table 1 reasonably efficient convex optimization problems can be found which allow one to solve the domain constrained optimization problem. That said, the optimization is often quite costly. For instance, the invariance based SDP constraints of Graepel and Herbrich (2004) are computationally tractable only if the number of observations is in the order of tens to hundreds, a far cry from requirements of massive datasets with thousands to millions of observations.

Even worse, the set S may not be finite and it may not be convex either. This means that the optimization problem, while convex, will not be able to incorporate S efficiently. We could, of course, circumscribe an ellipsoid for S by using a large γ to obtain a sufficient condition. This

approach, however, would typically lead to overly pessimistic classifiers. An alternative is constraint sampling, as proposed by (de Farias and Roy, 2004; Calafiore and Campi, 2004).

Let $f : \mathbb{R}^d \to \mathbb{R}$ and $c : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^l$ be convex functions, with $\Omega \subseteq \mathbb{R}^d$ being a closed convex set and $S \subseteq \mathbb{R}^l$. Consider the following optimization problem which is an instance of well known semi-infinite program

$$\underset{\theta \in \Omega}{\text{minimize}}\, f(\theta) \text{ subject to } c(\theta, x) \leq 0 \text{ for all } x \in S. \tag{22}$$

Depending on S the problem may have infinite number of constraints, and is in general intractable for arbitrary f and c. The constraint sampling approach for such problems proceeds by first imposing a probability distribution over S and then obtaining N independent observations, $x_1, \ldots, x_N$ from the set S by sampling. Finally one solves the finite convex optimization problem

$$\underset{\theta \in \Omega}{\text{minimize}}\, f(\theta) \text{ subject to } c(\theta, x_i) \leq 0 \text{ for all } 1 \leq i \leq N. \tag{23}$$

The idea is that by satisfying N constraints there is a high probability that an arbitrary constraint $c(x, \theta)$ is also satisfied. Let θ_N be the solution of (23). Note that since x_i are random variables θ_N, is also a random variable. The choice of N is given by a theorem due to Calafiore and Campi (2004).

Theorem 3 *Let $\varepsilon, \beta \in (0,1)$ and let $\theta \in \mathbb{R}^d$ be the decision vector then*

$$\Pr\{V(\theta_N) \leq \varepsilon\} \geq 1 - \beta \text{ where } V(\theta_N) = \Pr\{c(\theta_N, x) > 0 | x \in S\}$$

holds if

$$N \geq 2\left[d\varepsilon^{-1}\log\varepsilon^{-1} + \varepsilon^{-1}\log\beta^{-1} + d\right],$$

provided the set $\{x \in S | c(\theta_N, x) > 0\}$ is measurable.

Such a choice of N guarantees that the optimal solution θ_N of the sampled problem (23) is ε level feasible solution of the robust optimization problem (22) with high probability. Specializing this approach for the problem at hand would require drawing N independent observations from the set S_i, for each uncertain constraint, and replacing the SOCP constraint by N linear constraints of the form

$$y(w^\top x_i^j + b) \geq 1 \text{ for all } j \in \{1, \ldots N\}.$$

The choice of N is given by Theorem 3. Clearly the resulting problem is convex and has finite number of constraints. More importantly this makes the robust problem same as the standard SVM optimization problem but with more number of constraints.

In summary the advantage with the constraint sampling approach is one can still solve a robust problem by using a standard SVM solver instead of an SOCP. Another advantage is the approach easily carries over to arbitrary feature spaces. The downside of Theorem 3 is that N depends linearly on the *dimensionality* of w. This means that for nonparametric setting tighter bounds are required.[5]

5. Such bounds are subject to further work and will be reported separately.

5. Regression

Beyond classification the robust optimization approach can also be extended to regression. In this case one aims at finding a function $f : \mathcal{X} \to \mathcal{Y}$ such that some measure of deviation $c(e)$ between the observations and predictions, where $e(f(x),y) := f(x) - y$, is small. For instance we penalize

$$c(e) = \tfrac{1}{2}e^2 \qquad \text{LMS Regression } (l_2) \tag{24a}$$

$$c(e) = |e| \qquad \text{Median Regression } (l_1) \tag{24b}$$

$$c(e) = \max(0, |e| - \varepsilon) \qquad \varepsilon\text{-insensitive Regression} \tag{24c}$$

$$c(e) = \begin{cases} |e| - \frac{\sigma}{2} & \text{if } |e| \leq \sigma \\ \frac{1}{2\sigma}e^2 & \text{otherwise} \end{cases} \qquad \text{Huber's robust regression} \tag{24d}$$

The ℓ_1 and ℓ_2 losses are classical. The ε-insensitive loss was proposed by Vapnik et al. (1997), the robust loss is due to Huber (1982). Typically one does not minimize the empirical average over these losses directly but rather one minimizes the regularized risk which is composed of the empirical mean plus a penalty term on f controlling the capacity. See e.g. (Schölkopf and Smola, 2002) for further details.

Relatively little thought has been given so far to the problem when x may not be well determined. Bishop (1995) studies the case where x is noisy and he proves that this has a regularizing effect on the estimate. Our aim is complementary: we wish to find robust estimators which do not change significantly when x is only known approximately subject to some uncertainty. This occurs, e.g. when some coordinates of x are missing.

The basic tool for our approach are the Chebyshev and Gauss-Markov inequalities respectively to bound the first and second moment of $e(f(x),y)$. These inequalities are used to derive two SOCP formulations for designing robust estimators useful for regression with missing variables. Note that no distribution assumptions are made on the underlying uncertainty, except that the first and the second moments are available. Our strategy is similar to (Chandrasekaran et al., 1998; El Ghaoui and Lebret, 1997) where the worst case residual is limited in presence of bounded uncertainties.

5.1 Penalized Linear Regression and Support Vector Regression

For simplicity the main body of our derivation covers the linear setting. Extension to kernels is discussed in a later section Section 7. In penalized linear regression settings one assumes that there is a function

$$f(x) = \langle w, x \rangle + b, \tag{25}$$

which is used to minimize a regularized risk

$$\underset{w,b}{\text{minimize}} \sum_{i=1}^{n} c(e_i) \text{ subject to } \|w\| \leq W \text{ and } e_i = f(x_i) - y_i. \tag{26}$$

Here $W > 0$. As long as $c(e_i)$ is a convex function, the optimization problem (26) is a convex programming problem. More specifically, for the three loss functions of (24a) we obtain a quadratic program. For $c(e) = \frac{1}{2}e^2$ we obtain Gaussian Process regression estimators (Williams, 1998), in the second case we obtain nonparametric median estimates (Le et al., 2005), and finally $c(e) = \max(0, |e| - \varepsilon)$ yields ε-insensitive SV regression (Vapnik et al., 1997).

Eq. (26) is somewhat nonstandard insofar as the penalty on $\|w\|$ is imposed via the constraints rather than via a penalty in the objective directly. We do so in order to obtain second order cone programs for the robust formulation more easily without the need to dualize immediately. In the following part of the paper we will now seek means of bounding or estimating e_i subject to constraints on x_i.

5.2 Robust Formulations for Regression

We now discuss how to handle uncertainty in x_i. Assume that x_i is a random variable whose first two moments are known. Using the inequalities of Section 2.3 we derive two formulations which render estimates robust to the stochastic variations in x_i.

Denote by $\bar{x} := \mathbf{E}[x]$ the expected value of x. One option of ensuring robustness of the estimate is to require that the prediction errors are insensitive to the distribution over x. That is, we want that

$$\Pr_x\{|e(f(x),y) - e(f(\bar{x}),y)| \geq \theta\} \leq \eta, \tag{27}$$

for some confidence threshold θ and some probability η. We will refer to (27) as a "close to mean" (CTM) requirement. An alternative is to require that the residual $\xi(f(x),y)$ be small. We make use of a probabilistic version of the constraint $|e(f(x),y)| \leq \xi + \varepsilon$, that is equivalent to

$$\Pr_x\{|e(f(x),y)| \geq \xi + \varepsilon\} \leq \eta. \tag{28}$$

This is more geared towards good performance in terms of the loss function, as we require the estimator to be robust only in terms of deviations which lead to *larger* estimation error rather than requiring smoothness overall. We will refer to (28) as a "small residual" (SR) requirement. The following theorem shows how both quantities can be bounded by means of the Chebyshev inequality (6) and modified markov inequality (5).

Theorem 4 (Robust Residual Bounds) *Denote by $x \in \mathbb{R}^n$ a random variable with mean $\bar{x}$ and covariance matrix Σ. Then for $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ a sufficient condition for (27) is*

$$\left\|\Sigma^{\frac{1}{2}} w\right\| \leq \theta\sqrt{\eta}, \tag{29}$$

where $\Sigma^{\frac{1}{2}}$ is the matrix square root of Σ. Moreover, a sufficient condition for (28) is

$$\sqrt{w^\top \Sigma w + (\langle w,\bar{x}\rangle + b - y)^2} \leq (\xi + \varepsilon)\sqrt{\eta}. \tag{30}$$

Proof To prove the first claim note that for f as defined in (25), $\mathbf{E}(e(f(x),y)) = e(f(\bar{x}),y)$ which means that $e(f(x),y) - e(f(\bar{x}),y)$ is a zero-mean random variable whose variance is given by $w^\top \Sigma w$. This can be used with Chebyshev's inequaltiy (6) to bound

$$\Pr_x\{|e(f(x),y) - e(f(\bar{x}),y)| \geq \theta\} \leq \frac{w^\top \Sigma w}{\theta^2}. \tag{31}$$

Hence $w^\top \Sigma w \leq \theta^2 \eta$ is a sufficient condition for (27) to hold. Taking square roots yields (29). To prove the second part we need to compute the second order moment of $e(f(x),y)$. The latter is computed easily by the bias-variance decomposition as

$$\begin{aligned}\mathbf{E}\left[e(f(x),y)^2\right] &= \mathbf{E}\left[\left(e(f(x),y) - e(f(\bar{x}),y)\right)^2\right] + e(f(\bar{x}),y)^2 \\ &= w^\top \Sigma w + (\langle w,\bar{x}\rangle + b - y)^2.\end{aligned} \tag{32}$$

Using (5), we obtain a sufficient condition for (28)

$$w^\top \Sigma w + (\langle w, \bar{x}\rangle + b - y)^2 \leq (\xi + \varepsilon)^2 \eta. \tag{33}$$

As before, taking the square root yields (30). ■

5.3 Optimization Problems for Regression

The bounds obtained so far allow us to recast (26) into a robust optimization framework. The key is that we replace the equality constraint $e_i = f(x_i) - y_i$ by one of the two probabilistic constraints derived in the previous section. In the case of (27) this amounts to solving

$$\begin{aligned}
\underset{w,b,\theta}{\text{minimize}} \; & \sum_{i=1}^n c(e_i) + D\sum_{i=1}^n \theta_i && && \text{(34a)}\\
\text{subject to } & \|w\| \leq W \text{ and } \theta_i \geq 0 && \text{for all } 1 \leq i \leq n && \text{(34b)}\\
& \langle \bar{x_i}, w\rangle + b - y_i = e_i && \text{for all } 1 \leq i \leq n && \text{(34c)}\\
& \|\Sigma_i^{\frac{1}{2}} w\| \leq \theta_i \sqrt{\eta_i} && \text{for all } 1 \leq i \leq n, && \text{(34d)}
\end{aligned}$$

where (34d) arises from $\Pr_{x_i}\{|e(f(x_i), y_i) - e(f(\bar{x_i}), y_i)| \geq \theta_i\} \leq \eta_i$. Here D is a constant determining the degree of uncertainty that we are going to accept large deviations. Note that (34) is a *convex* optimization problem for all convex loss functions $c(e)$. This means that it constitutes a general robust version of the regularized linear regression problem and that all adjustments including the ν-trick can be used in this context. For the special case of ε-insensitive regression (34) specializes to an SOCP. Using the standard decomposition of the positive and negative branch of $f(x_i) - y_i$ into ξ_i and ξ_i^* Vapnik et al. (1997) we obtain

$$\begin{aligned}
\underset{w,b,\xi,\xi^*,\theta}{\text{minimize}} \; & \sum_{i=1}^n (\xi_i + \xi_i^*) + D\sum_{i=1}^n \theta_i && && \text{(35a)}\\
\text{subject to } & \|w\| \leq W \text{ and } \theta_i, \xi_i, \xi_i^* \geq 0 && \text{for all } 1 \leq i \leq n && \text{(35b)}\\
& \langle \bar{x_i}, w\rangle + b - y_i \leq \varepsilon + \xi_i \text{ and } y_i - \langle \bar{x_i}, w\rangle - b \leq \varepsilon + \xi_i^* && \text{for all } 1 \leq i \leq n && \text{(35c)}\\
& \|\Sigma_i^{\frac{1}{2}} w\| \leq \theta_i \sqrt{\eta_i} && \text{for all } 1 \leq i \leq n. && \text{(35d)}
\end{aligned}$$

In the same manner, we can use the bound (30) for (28) to obtain an optimization problem which minimizes the regression error directly. Note that (28) already allows for a margin ε in the regression error. Hence the optimization problem becomes

$$\begin{aligned}
\underset{w,b,\xi}{\text{minimize}} \; & \sum_{i=1}^n \xi_i && && \text{(36a)}\\
\text{subject to } & \|w\| \leq W \text{ and } \xi_i \geq 0 && \text{for all } 1 \leq i \leq n && \text{(36b)}\\
& \sqrt{w^\top \Sigma_i w + (\langle w, \bar{x_i}\rangle + b - y_i)^2} \leq (\xi_i + \varepsilon)\sqrt{\eta_i} && \text{for all } 1 \leq i \leq n. && \text{(36c)}
\end{aligned}$$

Note that (36) is an SOCP. In our experiments we will refer to (35) as the "close-to-mean" (CTM) formulation and to (36) as the "small-residual" (SR) formulation.

5.4 Geometrical Interpretation and Error Measures

The CTM formulation can be motivated by a similar geometrical interpretation to the one in the classification case, using an ellipsoid with center $\bar{x}$, shape and size determined by Σ and γ.

Theorem 5 *Assume that x_i is uniformly distributed in $\mathcal{E}(\bar{x}_i, \Sigma_i, \frac{1}{\sqrt{\eta_i}})$ and let f be defined by (25). In this case (35d) is a sufficient condition for the following requirement:*

$$|e(f(x_i),y) - e(f(\bar{x}_i),y)| \leq \theta_i \;\; \forall x_i \in \mathcal{E}_i \text{ where } \mathcal{E}_i := \mathcal{E}\left(\bar{x}_i, \Sigma_i, \eta_i^{-\frac{1}{2}}\right). \tag{37}$$

Proof Since $f(x) = \langle w,x\rangle + b$, left inequality in (37) amounts to $|\langle w,x_i\rangle - \langle w,\bar{x}_i\rangle| \leq \theta_i$. The inequality holds for all $x_i \in \mathcal{E}_i$ if $\max_{x_i\in\mathcal{E}_i} |\langle w,x_i\rangle - \langle w,\bar{x}_i\rangle| \leq \theta_i$. Application of Lemma 2 yields the claim. ■

A similar geometrical interpretation can be shown for SR. Motivated from this we define the following error measures.

Robustness Error: from the geometrical interpretation of CTM it is clear that $\gamma\|\Sigma^{\frac{1}{2}}w\|$ is the maximum possible difference between x and any other point in $\mathcal{E}(\bar{x},\Sigma,\gamma)$, since a small value of this quantity means smaller difference between $e(f(x_i),y_i))$ and $e(f(\bar{x_i}),y_i))$, we call $e_{\text{robust}}(\Sigma,\gamma)$ the *robustness error* measure for CTM

$$e_{\text{robust}}(\Sigma,\gamma) = \gamma\|\Sigma^{\frac{1}{2}}w\|. \tag{38}$$

Expected Residual: from (32) and (33) we can infer that SR attempts to bound the expectation of the square of the residual. We denote by $e_{\text{exp}}(\Sigma,\bar{x})$ an error measure for SR where,

$$e_{\text{exp}}(\bar{x},\Sigma) = \sqrt{w^\top\Sigma w + (e(f(\bar{x}),y))^2}. \tag{39}$$

Worst Case Error: since both CTM and SR are attempting to bound $w^\top\Sigma w$ and $e(f(\bar{x}_i),y_i)$ by minimizing a combination of the two and since the maximum of $|e(f(x),y)|$ over $\mathcal{E}(\bar{x},\Sigma,\gamma)$ is $|e(f(\bar{x}),y)| + \gamma\|\Sigma^{\frac{1}{2}}w\|$ (see Lemma 2) we would expect this worst case residual $w(\bar{x},\Sigma,\gamma)$ to be low for both CTM and SR. This measure is given by

$$e_{\text{worst}}(\bar{x},\Sigma,\gamma) = |e(f(\bar{x}),y)| + \gamma\|\Sigma^{\frac{1}{2}}w\|. \tag{40}$$

6. Robust Formulation For Missing Values

In this section we discuss how to apply the robust formulations to the problem of estimation with missing values. While we use a linear regression model to fill in the missing values, the linear assumption is not really necessary: as long as we have information on the first and second moments of the distribution we can use the robust programming formulation for estimation.

6.1 Classification

We begin by computing the sample mean and covariance for each class from the available observations, using a linear model and Expectation Maximization (EM) (Dempster et al., 1977) to take care of missing variables wherever appropriate:

Let (x,y) have parts x_m and x_a, corresponding to missing and available components respectively. With mean μ and covariance Σ for the class y and with decomposition

$$\mu = \begin{bmatrix} \mu_a \\ \mu_m \end{bmatrix} \text{ and } \Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{am} \\ \Sigma_{am}^\top & \Sigma_{mm} \end{bmatrix}, \tag{41}$$

we can now find the imputed means and covariances. They are given by

$$\mathbf{E}[x_m] = \mu_m + \Sigma_{ma}\Sigma_{aa}^{-1}(x_a - \mu_a) \tag{42}$$

$$\text{and } \mathbf{E}\left[x_m x_m^\top\right] - \mathbf{E}[x_m]\,\mathbf{E}[x_m]^\top = \Sigma_{mm} - \Sigma_{ma}\Sigma_{aa}^{-1}\Sigma_{ma}^\top. \tag{43}$$

In standard EM fashion one begins with initial estimates for mean and covariance, uses the latter to impute the missing values for the entire class of data and iterates by re-estimating mean and covariance until convergence.

Optimization Problem Without loss of generality, suppose that the patterns 1 to c are complete and that patterns $c+1$ to n have missing components. Using the above model we have the following robust formulation:

$$\underset{w,b,\xi}{\text{minimize}} \sum_{i=1}^{n} \xi_i \tag{44a}$$

$$\text{subject to } y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i \qquad \text{for all } 1 \leq i \leq c \tag{44b}$$

$$y_i\left(\langle w, \bar{x}_i\rangle + b\right) \geq 1 - \xi_i + \gamma_i \left\| \Sigma_i^{\frac{1}{2}} w \right\| \qquad \text{for all } c+1 \leq i \leq n \tag{44c}$$

$$\|w\| \leq W \;\text{ and }\; \xi_i \geq 0 \qquad \text{for all } 1 \leq i \leq n, \tag{44d}$$

where $\bar{x}_i$ denotes the pattern with the missing values filled in and

$$\Sigma_i = \begin{bmatrix} 0 & 0 \\ 0 & \Sigma_{mm} - \Sigma_{ma}\Sigma_{aa}^{-1}\Sigma_{am} \end{bmatrix}$$

according to the appropriate class labels. By appropriately choosing γ_i's, we can control the degree of robustness to uncertainty that arises out of imputation. The quantities γ_i's are defined only for the patterns with missing components.

Prediction After determining w and b by solving (44) we predict the label y of the pattern x by the following procedure.

1. If x has no missing values use it for step 4.
2. Fill in the missing values x_m in x using the parameters (mean and the covariance) of each class, call the resulting patterns x_+ and x_- corresponding to classes $+1$ and -1 respectively.
3. Find the distances d_+, d_- of the imputed patterns from the hyperplane, that is

   $$d_\pm := \left(w^\top x_\pm + b\right)\left(w^\top \Sigma_\pm w\right)^{-\frac{1}{2}}.$$

   Here $\Sigma_\pm$ are the covariance matrices of x_+ and x_-. These values tell which class gives a better fit for the imputed pattern. We choose that imputed sample which has higher distance from the hyperplane as the better fit: if $|d_+| > |d_-|$ use x_+, otherwise use x_- for step 4.
4. Calculate $y = \operatorname{sgn}\left(w^\top x + b\right)$.

6.2 Regression

As before we assume that the first c training samples are complete and the remaining training samples have missing values. After using the same linear model an imputation strategy as above we now propose to use the CTM and SR formulations to exploit the covariance information to design robust prediction functions for the missing values.

Once the missing values are filled in, it is straightforward to use our formulation. The CTM formulation for the missing values case takes the following form

$$\begin{aligned}
\underset{w,b,\theta,\xi,\xi^*}{\text{minimize}} \; & \sum_{i=1}^{n}(\xi_i+\xi_i^*)+D\sum_{i=c+1}^{n}\theta_i && && \text{(45a)}\\
\text{subject to } & \langle w,x_i\rangle+b-y_i\le\varepsilon+\xi_i \,,\, y_i-\langle w,x_i\rangle-b\le\varepsilon+\xi_i^* && \text{for all } 1\le i\le c && \text{(45b)}\\
& \langle w,\bar{x}_i\rangle+b-y_i\le\varepsilon+\xi_i \,,\, y_i-\langle w,\bar{x}_i\rangle-b\le\varepsilon+\xi_i^* && \text{for all } c+1\le i\le n && \text{(45c)}\\
& \left\|\Sigma_i^{\frac{1}{2}}w\right\|\le\theta_i\sqrt{\eta_i} && \text{for all } c+1\le i\le n && \text{(45d)}\\
& \theta_i\ge 0 \;\text{ for all } c+1\le i\le n \;\text{ and }\; \xi_i,\xi_i^*\ge 0 && \text{for all } 1\le i\le n && \text{(45e)}\\
& \|w\|\le W.
\end{aligned}$$

Only partially available data have the constraints (45d). As before, quantities θ_i's are defined only for patterns with missing components. A similar SR formulation could be easily obtained for the case of missing values:

$$\begin{aligned}
\underset{w,b,\xi,\xi^*}{\text{minimize}} \; & \sum_{i=1}^{c}(\xi_i+\xi_i^*)+\sum_{i=c+1}^{n}\xi_i\\
\text{subject to } & \langle w,x_i\rangle+b-y_i\le\varepsilon+\xi_i \;,\; y_i-\langle w,x_i\rangle-b\le\varepsilon+\xi_i^* && \text{for all } 1\le i\le c\\
& \sqrt{w^\top\Sigma_i w+(\langle w,\bar{x}_i\rangle+b-y_i)^2}\le(\varepsilon+\xi_i)\sqrt{\eta_i} && \text{for all } c+1\le i\le n\\
& \xi_i^*\ge 0 \text{ for all } 1\le i\le c \;\text{ and }\; \xi_i\ge 0 && \text{for all } 1\le i\le n\\
& \|w\|\le W.
\end{aligned}$$

7. Kernelized Robust Formulations

In this section we propose robust formulations for designing nonlinear classifiers by using kernel function. Note that a kernel function is a function $K:\Omega\times\Omega\to\mathbb{R}$, where K obeys the Mercer conditions (Mercer, 1909). We also extend these ideas to nonlinear regression functions.

7.1 Kernelized Formulations for Classification

The dual of the formulation (44), is given below (for a proof, please see Appendix A).

$$\underset{\lambda,\delta,\beta,u}{\text{maximize}} \sum_{i=1}^{n} \lambda_i - W\delta, \tag{47a}$$

$$\text{subject to } \sum_{i=1}^{n} \lambda_i y_i = 0, \tag{47b}$$

$$\|\sum_{i=1}^{c} \lambda_i y_i x_i + \sum_{i=c+1}^{n} \lambda_i y_i (\bar{x}_i + \gamma_i \Sigma_i^{\frac{1}{2}T} u_i)\| \leq \delta, \tag{47c}$$

$$\lambda_i + \beta_i = 1 \qquad \text{for all } 1 \leq i \leq n \tag{47d}$$

$$\|u_i\| \leq 1 \qquad \text{for all } c+1 \leq i \leq n \tag{47e}$$

$$\lambda_i, \beta_i, \delta \geq 0 \qquad \text{for all } 1 \leq i \leq n. \tag{47f}$$

The KKT conditions can be stated as (see Appendix A)

$$\sum_{i=1}^{c} \lambda_i y_i x_i + \sum_{i=c+1}^{n} \lambda_i y_i (\bar{x}_i + \gamma_i \Sigma_i^{\frac{1}{2}} u_i) = \delta u_{n+1} \tag{48a}$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0, \delta \geq 0 \tag{48b}$$

$$\lambda_i + \beta_i = 1,\ \beta_i \geq 0,\ \lambda_i \geq 0,\ \beta_i \lambda_i = 0 \qquad \text{for all } 1 \leq i \leq n \tag{48c}$$

$$\lambda_i (y_i (\langle w, x_i \rangle + b) - 1 + \xi_i) = 0 \qquad \text{for all } 1 \leq i \leq c \tag{48d}$$

$$\lambda_j (y_j (\langle w, \bar{x}_j \rangle + b) - 1 + \xi_j - \gamma_j (\Sigma_j^{\frac{1}{2}} u_j)) = 0 \qquad \text{for all } c+1 \leq j \leq n \tag{48e}$$

$$\delta(\langle w, u_{n+1} \rangle - W) = 0. \tag{48f}$$

The KKT conditions of the problem give some very interesting insights:

1. When $\gamma_i = 0$ $c+1 \leq i \leq n$ the method reduces to standard SVM expressed as an SOCP as it is evident from formulation (47).

2. When $\gamma_i \neq 0$ the problem is still similar to SVM but instead of a fixed pattern the solution chooses the vector $\bar{x}_i + \gamma_i \Sigma_i^{\frac{1}{2}} u_i$ from the uncertainty ellipsoid. Which vector is chosen depends on the value of u_i. Figure (2) has a simple scenario to show the effect of robustness on the optimal hyperplane.

3. The unit vector u_i maximizes $u_i^\top \Sigma_i^{\frac{1}{2}} w$ and hence u_i has the same direction as $\Sigma_i^{\frac{1}{2}} w$.

4. The unit vector u_{n+1} has the same direction as w. From (48a), for arbitrary data, one obtains $\delta > 0$, which implies $\langle w, u_{n+1} \rangle = W$ due to (48f). Substituting for u_{n+1} in (48a) gives the following expression for w,

$$w = \frac{W}{\delta} \left(\sum_{i=1}^{c} \lambda_i y_i x_i + \sum_{i=c+1}^{n} \lambda_i y_i \left(\bar{x}_i + \gamma_i \Sigma_i^{\frac{1}{2}} u_i \right) \right). \tag{49}$$

This expression for w is very similar to the expression obtained in the standard SVM. The vector w has been expressed as a combination of complete patterns and vectors from the uncertainty ellipsoid of the incomplete patterns.

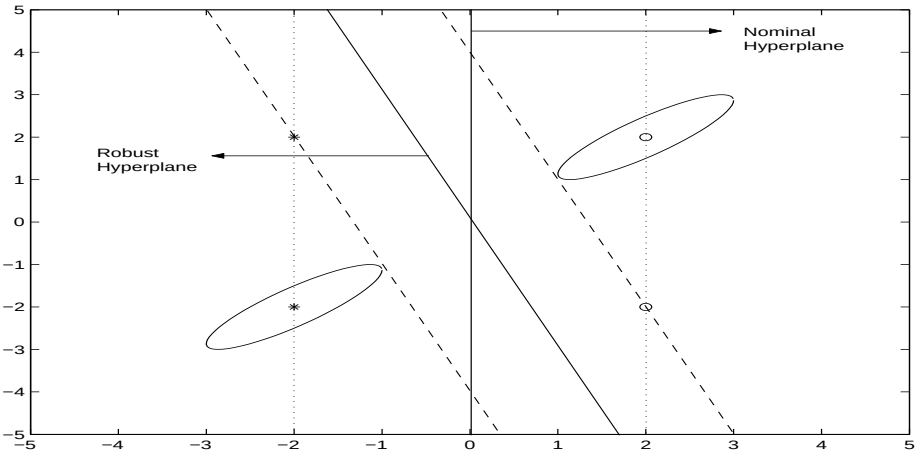

Figure 2: Circles and stars represent patterns belonging to the two classes. The ellipsoid around the pattern denotes the uncertainty ellipsoid. Its shape is controlled by the covariance matrix and the size by γ. The vertical solid line represents the optimal hyperplane obtained by nominal SVM while the thick dotted line represents the optimal hyperplane obtained by the robust classifier

Kernelized Formulation It is not simple to solve the dual (47) as a kernelized formulation. The difficulty arises from the fact that the constraint containing the dot products of the patterns (47c) involves terms such as $\left(\bar{x}_i+\gamma_i\Sigma_i^{\frac{1}{2}}u_i\right)^T\left(\bar{x}_j+\gamma_j\Sigma_j^{\frac{1}{2}}u_j\right)$ for some i and j. As u's are unknown, it is not possible to calculate the value of the kernel function directly. Hence we suggest a simple method to solve the problem from the primal itself.

When the shape of the uncertainty ellipsoid for a pattern with missing values is determined by the covariance matrix of the imputed values, any point in the ellipsoid is in the span of the patterns used in estimating the covariance matrix. This is because the eigenvectors of the covariance matrix span the entire ellipsoid. The eigenvectors of a covariance matrix are in the span of the patterns from which the covariance matrix is estimated. Since eigenvectors are in the span of the patterns and they span the entire ellipsoid, any vector in the ellipsoid is in the span of the patterns from which the covariance matrix is estimated.

The above fact and the equation to construct w from the dual variables (49) imply w is in the span of the imputed data (all the patterns: complete and the incomplete patterns with missing values imputed). Hence, $w=\sum_{i=1}^{c}\alpha_i x_i+\sum_{i=c+1}^{n}\alpha_i\bar{x}_i$.

Now, consider the constraint

$$y_i\left(\langle w,x_i\rangle+b\right)\geq 1-\xi_i.$$

It can be rewritten as,

$$y_i\left(\left\langle\left(\sum_{l=1}^{c}\alpha_l x_l+\sum_{l=c+1}^{n}\alpha_l\bar{x}_l\right),x_i\right\rangle+b\right)\geq 1-\xi_i.$$

We replace the dot product in the above equation by a kernel function to get

$$y_i\left(\left\langle \alpha, \tilde{K}(x_i)\right\rangle + b\right) \geq 1-\xi_i,$$

where $\tilde{K}(x_i)^T = [K(x_1,x_i),\ldots,K(x_c,x_i),K(\bar{x}_{c+1},x_i),\ldots,K(\bar{x}_n,x_i)]$ and $\alpha^\top = [\alpha_1,\ldots,\alpha_n]$. The observation x_i is either a complete pattern or a pattern with missing values filled in. Now, we consider the uncertainty in $\tilde{K}(x_i)$ to obtain the non-linear version of our formulation that can be solved easily. When we consider the uncertainty in $\tilde{K}(x_i)$ the probabilistic constraint takes the form

$$Pr\left(y_i\left(\left\langle \alpha, \tilde{K}(x_i)\right\rangle + b\right) \geq 1-\xi_i\right) \geq \kappa_i. \tag{50}$$

As in the original problem we now treat $\tilde{K}(x_i)$ as a random variable. The equation (50) has the same structure as the probabilistic constraint of Section 3. Following the same steps as in Section 3, it can be shown that the above probabilistic constraint is equivalent to

$$y_i\left(\left\langle \alpha, \tilde{K}(x_i)\right\rangle + b\right) \geq 1-\xi_i + \sqrt{\frac{\kappa_i}{1-\kappa_i}}\sqrt{\alpha^T\Sigma_i^k\alpha},$$

where Σ_i^k and $\tilde{K}(\bar{x}_i)$ are the covariance and the mean of $\tilde{K}(x_i)$ (in $\tilde{K}$-space). In view of this, the following is the non-linear version of the formulation:

$$\underset{\alpha,b,\xi}{\text{minimize}} \sum_{i=1}^{n}\xi_i \tag{51a}$$

$$\text{subject to } y_i\left(\left\langle \alpha, \tilde{K}(x_i)\right\rangle + b\right) \geq 1-\xi_i \qquad \text{for all } 1\leq i\leq c \tag{51b}$$

$$y_i\left(\left\langle \alpha, \tilde{K}(\bar{x}_j)\right\rangle + b\right) \geq 1-\xi_j+\gamma_j\left\|\Sigma_j^{k\frac{1}{2}}\alpha\right\| \qquad \text{for all } c+1\leq j\leq n \tag{51c}$$

$$\|\alpha\| \leq W \;\; \xi_i \geq 0 \qquad \text{for all } 1\leq i\leq n. \tag{51d}$$

The constraint (51d) follows from the fact that we are now doing linear classification in $\tilde{K}$-space. The constraint is similar to the constraint $\|w\| \leq W$ which we had in the linear versions.

Estimation of Parameters A point to be noted here is that Σ_j^k defines the uncertainty in $\tilde{K}(x_j)$. In the original lower dimensional space we had a closed form formula to estimate the covariance for patterns with missing values. However, now we face a situation where we need to estimate the covariance in $\tilde{K}$-space. A simple way of doing this is to assume spherical uncertainty in $\tilde{K}$-space. Another way of doing this is by a nearest neighbour based estimation. To estimate the covariance of $\tilde{K}(x_i)$, we first find out k nearest neighbours of x_i and then we estimate the covariance from $\tilde{K}(x_{i_1}),\ldots,\tilde{K}(x_{i_k})$ where $x_{i_1},\ldots,x_{i_k}$ are the nearest neighbours of x_i.

It is straight forward to extend this more general result (51) to the missing value problem following the same steps as in (6).

Classification Once α's are found, given a test pattern t its class is predicted in the following way: If the pattern is incomplete, it is first imputed using the way it was done during training. However, this can be done in two ways, one corresponding to each class as the class is unknown for the pattern. In that case the distance of each imputed pattern from the hyperplane is computed from

$$h_1 = \frac{\alpha^T\tilde{K}(t)+b}{\sqrt{\alpha^T\Sigma_1\alpha}} \text{ and } h_2 = \frac{\alpha^T\tilde{K}(t)+b}{\sqrt{\alpha^T\Sigma_2\alpha}},$$

where Σ_1 and Σ_2 are the covariances obtained by the same strategy as during training. Higher of the above two is selected as it gives a better fit for the pattern. The prediction for the pattern is the prediction of its centroid (i.e. the prediction for the centroid which gives a better fit). Let $h = \max(|h_1|, |h_2|)$, if $h = |h_1|$ then $y = \text{sgn}(h_1)$ else $y = \text{sgn}(h_2)$ where y is the prediction for the pattern t. In case the pattern is complete, there is no ambiguity we can give $\text{sgn}(\alpha^T \tilde{K}(t) + b)$ as the prediction.

7.2 Kernelized Robust Formulations for Regressions

As discussed for the case of classification we derive nonlinear regressions functions by using the $\tilde{K}$. We fit a hyperplane (α, b) in the $\tilde{K}$ where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$. Whenever x is a random variable we consider $\tilde{K}(x)$ as a random variable with mean $\tilde{K}(\bar{x})$ and with either unit covariance or a covariance estimated from nearest neighbours in the $\tilde{K}$-space. Instead of finding (w, b) we resort to finding (α, b) where α plays the role of w but in the $\tilde{K}$-space. Essentially, we just have to replace w by α and x_i by $\tilde{K}x_i$ and the covariance by the estimate covariance in the $\tilde{K}$-space. Given these facts, we get the following kernelized version of the Close To Mean formulation:

$$\begin{aligned}
\underset{\alpha,b,\theta,\xi,\xi^*}{\text{minimize}} & \sum_{i=1}^{n}(\xi_i + \xi_i^*) + D\sum_{i=1}^{n}\theta_i \\
\text{subject to } & \left\langle \alpha, \tilde{K}(\bar{x}_i)\right\rangle + b - y_i \le \varepsilon + \xi_i && \text{for all } 1 \le i \le n \\
& y_i - \left\langle \alpha, \tilde{K}(\bar{x}_i)\right\rangle - b \le \varepsilon + \xi_i^* && \text{for all } 1 \le i \le n \\
& \sqrt{\alpha^\top \Sigma_i^k \alpha} \le \theta_i \sqrt{\eta_i} && \text{for all } 1 \le i \le n \\
& \|\alpha\| \le W \text{ and } \theta_i, \xi_i, \xi_i^* \ge 0 && \text{for all } 1 \le i \le n.
\end{aligned}$$

Similarly, the kernelized version of formulation SR is given by,

$$\begin{aligned}
\underset{\alpha,b,\xi}{\text{minimize}} & \sum_{i=1}^{n}\xi_i \\
\text{subject to } & \sqrt{\alpha^\top \Sigma_i^k \alpha + (\left\langle \alpha, \tilde{K}(\bar{x}_i)\right\rangle + b - y_i)^2} \le (\varepsilon + \xi_i)\sqrt{\eta_i} && \text{for all } 1 \le i \le n \\
& \|\alpha\| \le W \text{ and } \xi_i \ge 0 && \text{for all } 1 \le i \le n.
\end{aligned}$$

In the above formulations, Σ_i^k is the estimate covariance in the $\tilde{K}$-space. If the patterns 1 through c are complete and the patterns $c+1$ through n have missing values, then assuming $\eta_i = 1$ and $\Sigma_i^k = 0$ for i from 1 through c, would make the above formulations directly applicable to the case.

8. Experiments

In this section we empirically test the derived formulations for both classification and regression problems which have missing values in the observations. In all the cases interior point method was used to solve SOCP using the commercially avilable Mosek solver.

8.1 Classification

We consider the classification case first. Consider a binary classification problem with training data having missing values. The missing values are filled in by imputation and subsequently a

SVM classifier was trained on the complete data to obtain the *nominal classifier*. We compared the proposed formulations with the nominal classifiers by performing numerical experiments on real life data bench mark datasets. We also use a non-linear separable data set to show that the kernelized version works when the linear version breaks down. In our formulations we will assume that $\gamma_j = \gamma$.

For evaluating the results of robust classifier we used the worst case error and the expected error along with the actual error. A test pattern with no missing values can be directly classified. In case it has missing values, we first impute the missing values and then classify the pattern. We refer to the error on a set of patterns using this approach the actual error.

We first consider the problem of classifying OCR data where missing values can occur more frequently. Specifically we consider the classification problem between the two digits '3' and '8'. We have used the UCI (Blake and Merz, 1998) OCR data set, A data set is generated by deleting 75% of the pixels from 50% of the training patterns. Missing values were then imputed using linear regression. We trained a SVM on this imputed data, to obtain the nominal classifier. This was compared with the robust classifier trained with different values of γ, corresponding to different degrees of confidence as stated in (11).

The error rates of the classifiers were obtained on the test data set by randomly deleting 75% of the pixels from each pattern. We then repeated 10 such iterations and obtained the average error rates. Figure 3 shows some of the digits that were misclassified by the nominal classifier but were correctly classified by the robust classifier. The effectiveness of our formulation is evident from these images. With only partial pixels available, our formulation did better than the nominal classifier. Figure 4 show the different error rates obtained on this OCR data set. In all the three measures, the robust classifier outperformed the nominal classifier.

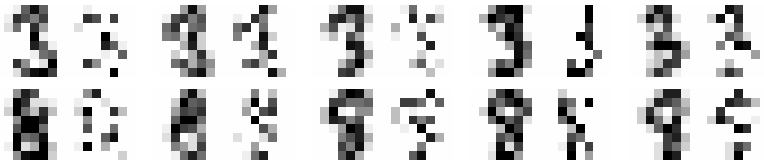

Figure 3: In all images the left image shows a complete digit, the right image shows the digit after randomly deleting 75% of the pixels. The first five are '3' while the next five are '8'.

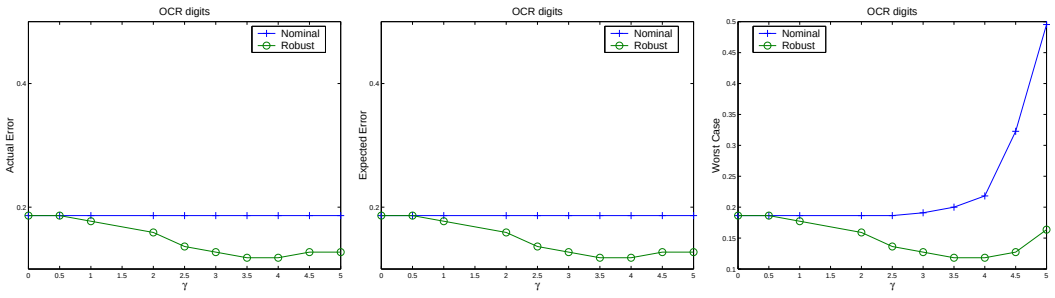

Figure 4: Error rates against γ with linear classifier on the OCR data.

Here we report the error rates using three measures we defined for three other UCI data sets (Blake and Merz (1998)), Heart, Ionosphere and Sonar. Linear version of our formulation was used. Experiments were done with low noise (50% patterns with missing values) and high noise (90% patterns with missing values). The data sets were divided in the ratio 9:1, the larger set was used for training the nominal and robust classifiers while the smaller set was used as test data set. 50% of the feature values (chosen at random) were deleted from 50% of the training patterns (in the low noise case) and 90% of the training patterns (in the high noise case). Linear regression based model was used to fill in the missing values. Nominal classifier and robust classifiers with different values of γ were trained using each such data set. Error rates were obtained for the test data set after deleting 50% of the feature values from each test pattern. The error rates reported here are over ten such randomized iterations.

The error rates as a function of γ are plotted in Figures 5,6 and 7. In case of actual error, the plots also show a horizontal line labeled 'clean' which is the error rate on the actual data set without any missing values. In this case, we did not delete any feature values from the data set. Nominal classifiers were trained and testing was also done on complete test samples. Our aim was to see how close our robust classifier could get near the error rates obtained using the complete data set.

It can be seen that the robust classifier, with suitable amount of robustness comes very close to the error rates on the clean data set. Amongst the three error measures the worst case error, the last column of Figure 7, brings out the advantage of the robust classifier over the nominal classifier. Clearly with increasing γ the robust formulation gives dividends over the nominal classifier.

We also did experiments to compare the kernelized version of the formulation over the linear formulation. For this purpose, we generated a dataset as follows. The positive class was obtained by generating uniformly distributed points in a hypershpere in $\mathbb{R}^5$ of unit radius centered at the origin. The negative class was obtained by generating uniformly distributed points in a annular band of thickness one, with the inner radius two, centered around the origin. In summary

$$y = \begin{cases} 1 & \|x\| \leq 1 \\ -1 & 2 \leq \|x\| \leq 3, \end{cases}$$

where $x \in \mathbb{R}^5$. An illustration of how such a dataset looks in two dimensions is given in the left of Figure 8. Hundred patterns were generated for each class. The data set was divided in the ratio 9:1. The larger part was used for training, the smaller part for testing. Three randomly chosen values were deleted from the training data set. The missing values were filled in using linear regression based strategy. We trained a classifier for different values of γ. Actual Error was found out for both the kernelized version and the linear version of the formulation. The results reported here are over ten such randomized runs. Gaussian kernel $(K(x,y) = \exp(-q\|x-y\|^2))$ was used in the case of kernelized formulation. The parameter q was chosen by cross validation. Spherical uncertainty was assumed in $\tilde{K}$-space for samples with missing values in case of kernelized robust formulations.

Figure 8 shows actual error rates with linear nominal, linear robust, kernelized nominal and kernelized robust. It can be seen that the linear classifier has broken down, while the kernelized classifier has managed a smaller error rate. It can also be observed that the robust kernelized classifier has the least error rate.

8.2 Regression

Given a regression problem with training data having missing values in the observations we obtained the *nominal regression* function by training a Support Vector Regression(SVR) formulation over the

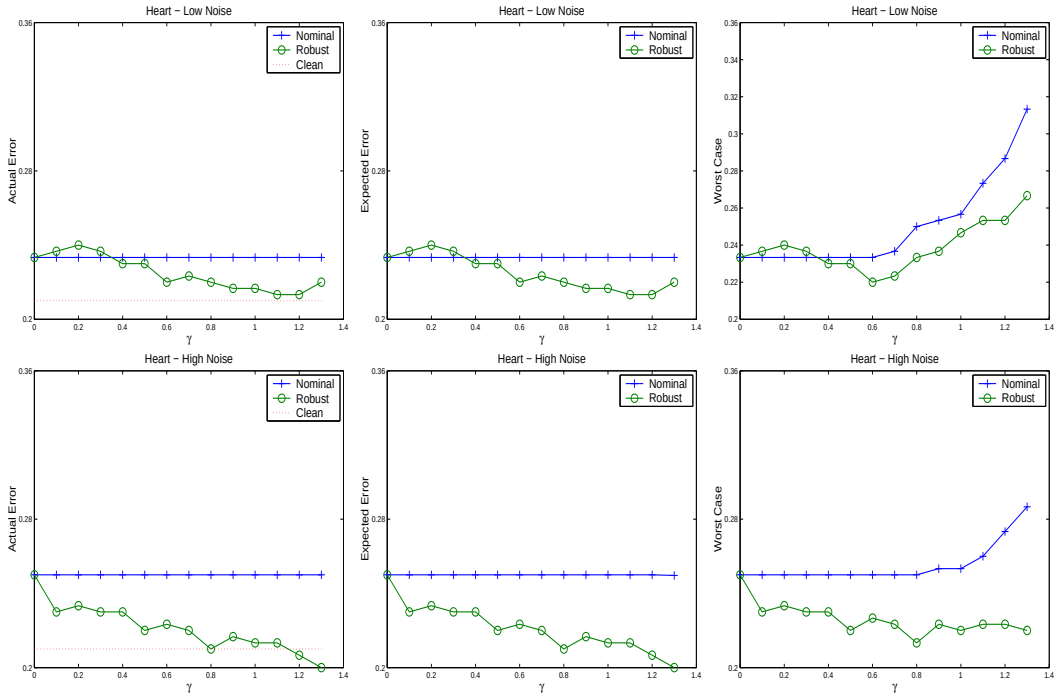

Figure 5: Error rates as a function of γ for Heart. Patterns in the top row contained 50% missing variables, even ones 90%. From left to right — actual error, expected error, and worst case error.

imputed data. The obtained regression function will be called the nominal SVR. In this section we compare our formulations with nominal SVR on a toy dataset and one real world dataset in the linear setting. We also compared the kernelized formulations with the linear formulations.

The first set of results is on a toy data set consisting of 150 observations. Each observation consisted (y,x) pair where

$$y = w^\top x + b, \quad w^\top = [1,2,3,4,5], \quad b = -7.$$

Patterns x were generated from a Gaussian distribution with mean, $\mu = 0$, and randomly chosen covariance matrix, Σ. The results are reported for the following choice of Σ:

$$\begin{bmatrix} 0.1872 & 0.1744 & 0.0349 & -0.3313 & -0.2790 \\ 0.1744 & 0.4488 & 0.0698 & -0.6627 & -0.5580 \\ 0.0349 & 0.0698 & 0.1140 & -0.1325 & -0.1116 \\ -0.3313 & -0.6627 & -0.1325 & 1.3591 & 1.0603 \\ -0.2790 & -0.5580 & -0.1116 & 1.0603 & 0.9929 \end{bmatrix}.$$

Missing values were introduced by randomly choosing 50% of the examples and deleting 2 of the entries example selected at random for each chosen example. The data was divided in the ratio 9:1, the larger one was used for training and the smaller one was used for testing. The results reported here are the average over ten such randomly partitioned training and test data. After imputing

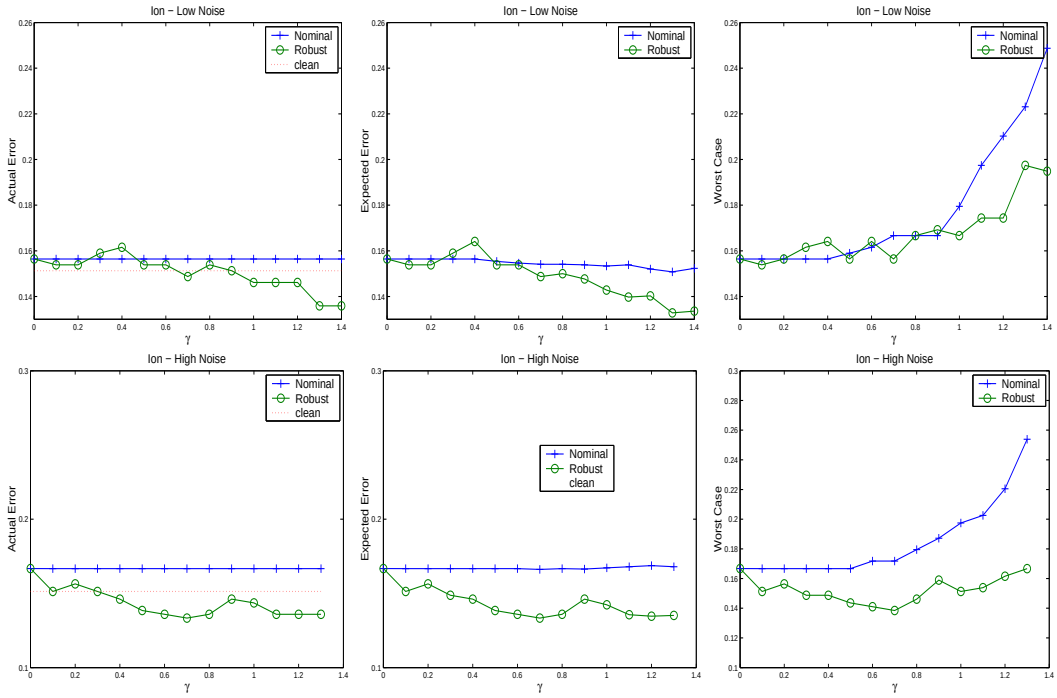

Figure 6: Error rates as a function of γ for Ionosphere. Patterns in the top row contained 50% missing variables, even ones 90%. From left to right — actual error, expected error, and worst case error.

the missing values using a linear regression model, training data was used with different formulations. The first row of Figure 9 shows robustness error (38), worst case error (40) for CTM and expected residual (39) and worst case error (40) for SR. The second row gives the results on UCI Blake and Merz (1998) boston data set with the same test methodology. The performance of our formulation over nominal regression is evident.

To validate our kernelized formulation, 150 samples in $\mathbb{R}^5$ were randomly generated as in the above case. For each x, the output is given by $y = c^T\phi(x) - c_0$, see footnote.[6] The mapping $\phi(x)$ is such that a hyperplane in $\mathbb{R}^{15}$ is actually a quadratic curve in $\mathbb{R}^5$. Randomly generated c and c_0 were used in this mapping. 40% of the values were deleted at random from 50% and 20% of the training samples for CTM and SR, they were filled in using the linear regression model. A Gaussian kernel $K(a,b) = exp(-\gamma\|a-b\|^2)$ with kernel parameter $\gamma = 0.1$ was used. Figure 10 shows the test errors per sample on 10 runs with different randomly deleted values. Test error is the error rate on a test set with missing values filled in. Essentially, we calculate $\sum_{i=1}^{n} e(f(\bar{x}_i, y_i))$ for all the test samples where the missing values are filled in using the training data set parameters using a linear

6. Let $x = [x_1, x_2, \ldots, x_5]$. The mapping $\phi : \mathbb{R}^5 \to \mathbb{R}^{15}$ is defined by

$$\phi(x) = [x_1^2 x_2^2\ x_3^2\ \ x_4^2\ x_5^2\ \ \sqrt{2}x_1x_2\ \sqrt{2}x_1x_3\ \sqrt{2}x_1x_4\ \sqrt{2}x_1x_5\ \sqrt{2}x_2x_3\ \sqrt{2}x_2x_4\ \sqrt{2}x_2x_5\ \sqrt{2}x_3x_4\ \sqrt{2}x_3x_5\ \sqrt{2}x_4x_5]^\top.$$

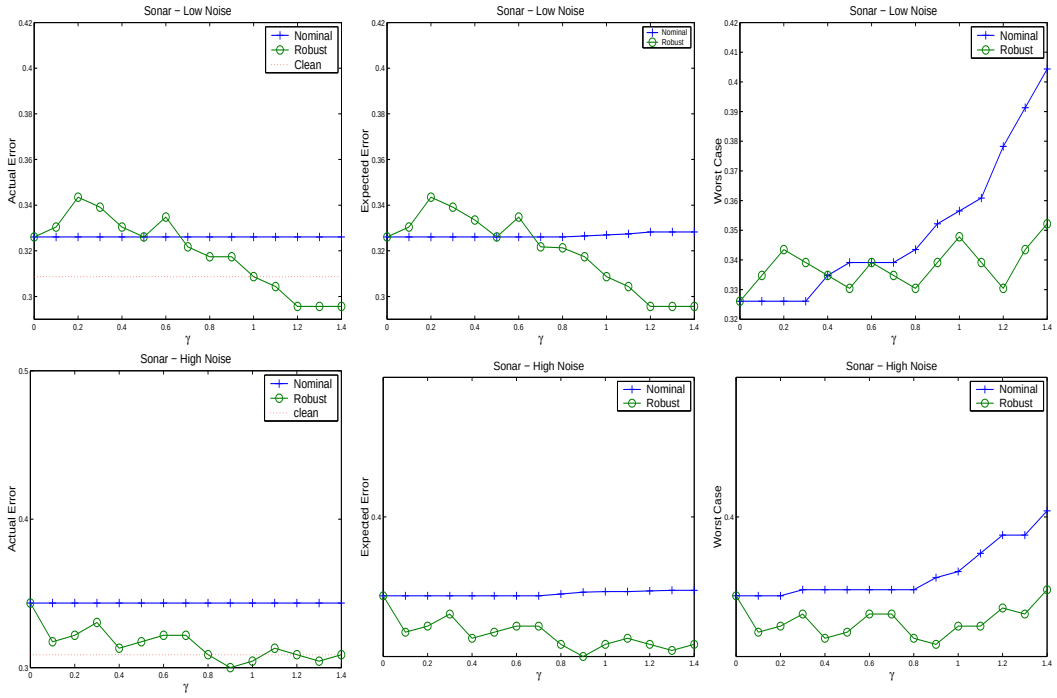

Figure 7: Error rates as a function of γ for Sonar. Patterns in the top row contained 50% missing variables, even ones 90%. From left to right — actual error, expected error, and worst case error.

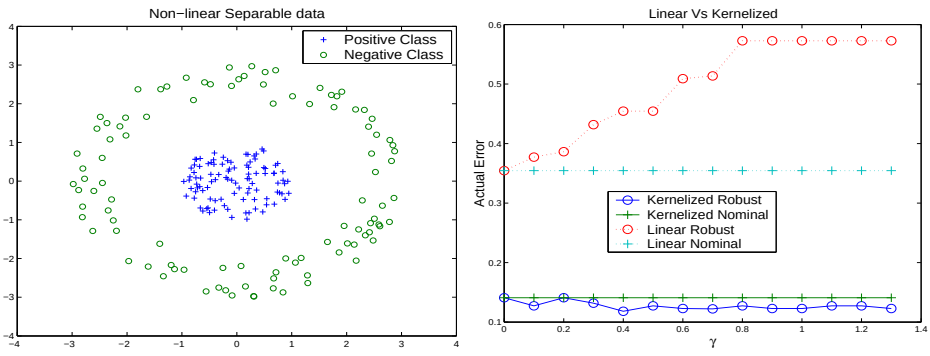

Figure 8: The left figure shows how the data set looks in two dimensions, the right figure gives the actual error rate for linear and kernelized formulations for the robust and nominal cases.

regression model. Essentially it is the absolute residual for imputed mean test data. The figures show that the kernelized version of the robust formulation does a better job than the linear version when the underlying function is non-linear.

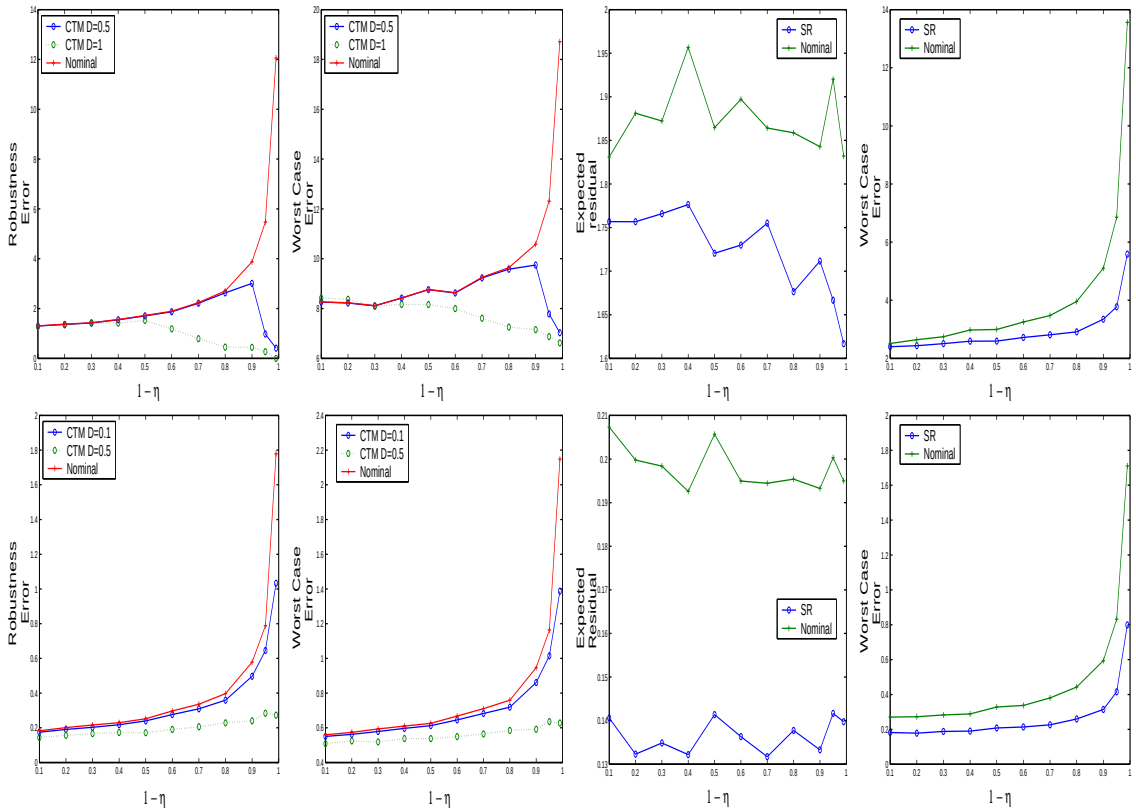

Figure 9: Top row — toy data set, Bottom row — Boston Housing estimation problem; From left to right: robustness (CTM), worst case error (CTM), expected residual (SR), and worst case error (SR). All graphs describe the error as a function of the robustness η.

9. Conclusions

In this paper we have proposed SOCP formulations for designing robust linear prediction functions which are capable of tackling uncertainty in the patterns both in classification and regression setting. The formulations are applicable to any uncertainty distribution provided the first two moments are computable. When applied to the missing variables problem the formulations outperform the imputation based classifiers and regression functions. We have also proposed a way to design nonlinear prediction functions by using regression setting.

The robustness in the context of classification can be geometrically interpreted as requiring that all points in the ellipsoid occur on one side of the hyperplane. Instead of having an ellipsoidal uncertainty one can have situations where the uncertainty is described by arbitrary sets. The constraint sampling approaches can serve as useful alternatives for such problems. Future work will consist in examining this approach for the problem at hand.

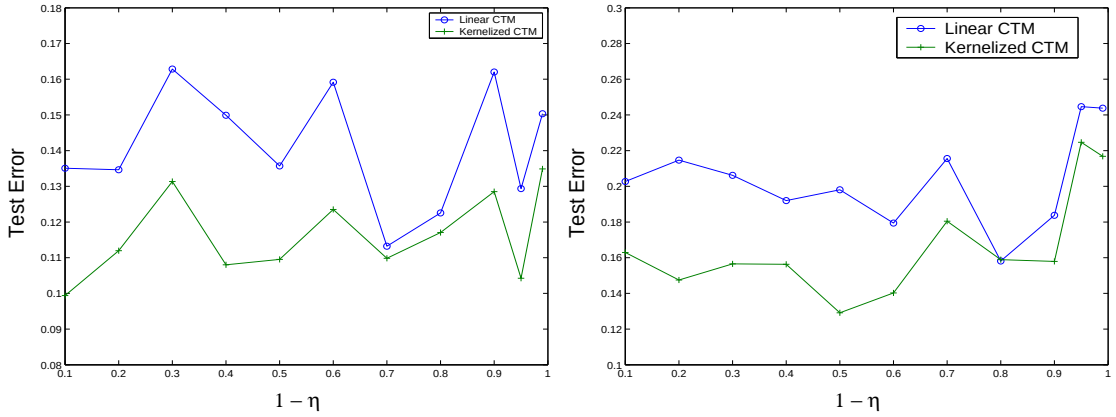

Figure 10: Linear vs. nonlinear regression. Left: CTM formulation, right: SR formulation.

Acknowledgments

CB was partly funded by MHRD (Grant number F26-11/2004). National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council. AS was supported by grants of the ARC. We thank Laurent ElGhaoui, Michael Jordan, Gunnar Rätsch, and Frederik Schaffalitzky for helpful discussions and comments.

Appendix A. Dual of the SOCP

The Lagrangian of (44) is given by

$$\begin{aligned}\mathcal{L}(w,\xi,b,\lambda,\beta,\delta) = &\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\beta_i\xi_i - \sum_{i=1}^{c}\lambda_i\left(y_i\left(w^T x_i + b\right) - 1 + \xi_i\right) \\ &- \sum_{j=c+1}^{n}\lambda_j\left(y_j(w^T\bar{x}_j + b) - 1 + \xi_j - \gamma_j\|\Sigma_j^{\frac{1}{2}}w\|\right) + \delta\left(\|w\| - W\right) \qquad (54)\\ &\beta_i,\lambda_i,\delta \geq 0.\end{aligned}$$

Recall that for any $x \in \mathbb{R}^n$ the relationship $\|x\|_2 = \max_{\|x\|\leq 1} x^\top y$ holds. This can be used to handle terms like $\left\|\Sigma_j^{\frac{1}{2}}w\right\|$ and $\|w\|$ leading to a modified Lagrangian given as follows

$$\begin{aligned}\mathcal{L}_1(w,\xi,b,\lambda,\beta,\delta,u) = &\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\beta_i\xi_i - \sum_{i=1}^{c}\lambda_i\left(y_i\left(w^T x_i + b\right) - 1 + \xi_i\right) \\ &- \sum_{j=c+1}^{n}\lambda_j\left(y_j(w^T\bar{x}_j + b) - 1 + \xi_j - \gamma_j\left(\Sigma_j^{\frac{1}{2}}w\right)^T u_j\right) + \delta\left(w^T u_{n+1} - W\right).\end{aligned} \qquad (55)$$

The Lagrangian $\mathcal{L}_1$ has the same optimal value as $\mathcal{L}$ when maximized with respect to u's subject to the constraints $\|u_i\| \leq 1$ for all $c+1 \leq i \leq n+1$. Note that the u's are defined only for patterns with

missing values and u_{n+1} is defined for the constraint $\|w\| \leq W$. Therefore

$$\mathcal{L}_1(w,\xi,b,\lambda,\beta,\delta) = \max_u \mathcal{L}(w,\xi,b,\lambda,\beta,\delta,u) \text{ subject to } \|u_i\| \leq 1 \text{ for all } i \in \{c+1,\ldots n+1\}.$$

By definition, solving (44) is equivalent to finding the saddle-point of the Lagrangian $\mathcal{L}_1$. By virtue of the above reasoning and due to convexity we obtain

$$\underset{w,b,\xi}{\text{minimize}} \underset{\lambda,\delta,\beta}{\text{maximize}} \mathcal{L}(w,\xi,b,\lambda,\beta,\delta) \tag{56a}$$

$$= \underset{w,b,\xi}{\text{minimize}} \underset{\lambda,\delta,\beta,\|u\|\leq 1}{\text{maximize}} \mathcal{L}_1(w,\xi,b,\lambda,\beta,\delta,u) \tag{56b}$$

$$= \underset{\lambda,\delta,\beta,\|u\|\leq 1}{\text{maximize}} \underset{w,b,\xi}{\text{minimize}} \mathcal{L}_1(w,\xi,b,\lambda,\beta,\delta,u). \tag{56c}$$

Eq (56c) now enables us to eliminate the primal variables to give the dual. Taking partial derivatives of $\mathcal{L}$ with respect to $w, b,$ and ξ yields

$$\partial_w \mathcal{L}(w,\xi,b,\lambda,\beta,\delta,u) = -\sum_{i=1}^{c} \lambda_i y_i x_i - \sum_{j=c+1}^{n} \lambda_j \left(y_j \bar{x}_j - \gamma_i {\Sigma_j}^{\frac{1}{2}T} u_j \right) + \delta u_{n+1} \tag{57a}$$

$$\partial_{\xi_i} \mathcal{L}(w,\xi,b,\lambda,\beta,\delta,u) = 1 - \lambda_i - \beta_i \tag{57b}$$

$$\partial_b \mathcal{L}(w,\xi,b,\lambda,\beta,\delta,u) = \sum_{i=1}^{n} \lambda_i y_i. \tag{57c}$$

Changing the sign of u_j for $c+1 \leq i \leq n$ does not matter since the optimal value of maximization of both $w^\top u_j$ and $-w^\top u_j$ over $\|u_j\| \leq 1$ are the same. Substituting $-u_j$ in (57a) by $y_j u_j$ and then equating (57a) , (57b) and (57c) to zero gives

$$\sum_{i=1}^{c} \lambda_i y_i x_i + \sum_{j=c+1}^{n} \lambda y_j \left(\bar{x}_j + \gamma_i {\Sigma_j}^{\frac{1}{2}T} u_j \right) = \delta u_{n+1} \tag{58a}$$

$$1 - \lambda_i - \beta_i = 0 \tag{58b}$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0. \tag{58c}$$

Substituting (58a), (58b) and (58c) in (55) subject to the relevant constraints yields the dual stated as follows

$$\underset{u,\lambda,\beta,\delta}{\text{maximize}} \sum_{i=1}^{n} \lambda_i - W\delta \tag{59a}$$

$$\text{subject to } \sum_{i=1}^{n} \lambda_i y_i = 0 \tag{59b}$$

$$\sum_{i=1}^{c} \lambda_i y_i x_i + \sum_{j=c+1}^{n} \lambda_j y_j \left(\bar{x}_j + \gamma_i {\Sigma_j}^{\frac{1}{2}T} u_j \right) = \delta u_{n+1} \tag{59c}$$

$$\lambda_i + \beta_i = 1 \qquad \text{for all } 1 \leq i \leq n \tag{59d}$$

$$\|u_i\| \leq 1 \qquad \text{for all } c+1 \leq i \leq n+1 \tag{59e}$$

$$\lambda_i, \beta_i, \delta \geq 0 \qquad \text{for all } 1 \leq i \leq n. \tag{59f}$$

For arbitrary data $\delta > 0$, which when plugged into (58a), gives

$$u_{n+1} = \frac{\sum_{i=1}^{c} \lambda_i y_i x_i + \sum_{j=c+1}^{n} \lambda_j y_j \left(\overline{x}_j + \gamma_i \Sigma_j{}^{\frac{1}{2}T} u_j \right)}{\delta}$$

and hence the dual (47) follows.

References

A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805, 1998.

A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001.

K. P. Bennett and O. L. Mangasarian. Multicategory separation via linear programming. *Optimization Methods and Software*, 3:27 – 39, 1993.

C. Bhattacharyya, L. R. Grate, M. I. Jordan, L. El Ghaoui, and Saira I. Mian. Robust sparse hyperplane classifiers: application to uncertain molecular profiling data. *Journal of Computational Biology*, 11(6):1073 – 1089, 2004a.

C. Bhattacharyya, K. S. Pannagadatta, and A. J. Smola. A second order cone programming formulation for classifying missing data. In *Advances in Neural Information Processing Systems (NIPS 17)*, 2004b.

J. Bi and T. Zhang. Support vector classification with input data uncertainty. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, 2004.

C. M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7:108 – 116, 1995.

C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

G. Calafiore and M. C. Campi. The scenario approach to robust control design. Technical report, Universita di Brescia, 2004. submitted.

S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed. Parameter Estimation in the Presence of Bounded Data Uncertainties. *SIAM J. Matrix Anal. Appl.*, 19(1):235–252, 1998.

M. Collins. Discriminative training methods for hidden markov models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2002.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

D. Pucci de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society B*, 39(1):1 – 22, 1977.

L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.*, 18(4):1035–1064, 1997.

G. Fung, O. L. Mangasarian, and J. W. Shavlik. Knowledge-based support vector machine classifiers. In *Advances in Neural Information Processing Systems 15*, volume 15. MIT Press, 2002.

T. Graepel and R. Herbrich. Invariant pattern recognition by semidefinite programming machines. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

P. J. Huber. *Robust statistics*. John Wiley, 1982.

G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharrya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

Q. V. Le, T. Sears, and A. J. Smola. Nonparametric quantile regression. Technical report, National ICT Australia, June 2005. Available at http://sml.nicta.com.au/∼quoc.le.

M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1 - 3):193 – 228, 1998.

A. W. Marshall and I. Olkin. Multivariate chebyshev inequalities. *Annals of Mathematical Statistics*, 31(4):1001–1014, 1960.

J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, A 209:415 – 446, 1909.

Y. Nesterov and A. Nemirovskii. *Interior Point Algorithms in Convex Programming*. Number 13 in Studies in Applied Mathematics. SIAM, Philadelphia, 1993.

G. Rätsch, S. Mika, and A. J. Smola. Adapting codes and embeddings for polychotomies. In *Neural Information Processing Systems*, volume 15. MIT Press, 2002.

T. Schneider. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate*, 14:853 – 871, 2001.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11/12(1 - 4):625 – 653, 1999.

B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, 2003.

V. Vapnik, S. Golowich, and A. J. Smola. Support vector method for function approximation, regression estimation, and signal processing. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 281 – 287, Cambridge, MA, 1997. MIT Press.

C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning and Inference in Graphical Models*, pages 599 – 621. Kluwer Academic, 1998.