Interval Data Classification under Partial Information: A Chance-Constraint Approach

Sahely Bhadra[1], J. Saketha Nath[2], Aharon Ben-Tal[2], and Chiranjib Bhattacharyya[1]

[1] Dept. of Computer Science and Automation,
Indian Institute of Science, Bangalore, INDIA.
sahely,chiru@csa.iisc.ernet.in
[2] Faculty of Industrial Engg. and Management,
Technion, Haifa, ISRAEL.
saketh@tx.technion.ac.il,abental@ie.technion.ac.il

Abstract. This paper presents a novel methodology for constructing maximum-margin classifiers which are robust to interval-valued uncertainty in examples. The idea is to employ chance-constraints which ensure that the uncertain examples are classified correctly with high probability. The key novelty is in employing Bernstein bounding schemes to relax the resulting chance-constrained program as a convex second order cone program. The Bernstein based relaxations presented in the paper require the knowledge of support and mean of the uncertain examples alone and make no assumptions on distributions regarding the underlying uncertainty. Classifiers built using the proposed methodology model interval-valued uncertainty in a less conservative fashion and hence are expected to generalize better than existing methods. Experimental results on synthetic and real-world datasets show that the proposed classifiers are better equipped to handle interval-valued uncertainty than state-of-the-art.

1 Introduction

In the recent past there has been a growing interest in analysis of interval-valued data in the learning community [1, 2]. In many real world problems it is not possible to describe the data by a precise value but intervals may be a more proper description. For example, in the case of cancer diagnosis, a tumorous tissue is analyzed and features are computed for each cell nucleus. However, since the features vary among cells of a tissue, usually, the mean and worst-case (minimum/maximum) feature values of the tissues are considered[3]. Also in the case of gene-expression data like micro-array, since the experiments are usually noisy and data is prone to be erroneous, data for a number of replicates of the same experiment are provided. Past research has shown that handling uncertainty in

[3] Examples are the Wisconsin breast cancer diagnostic/prognostic datasets available at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

such applications by the representation as interval data leads to accurate learning algorithms [3, 1]. Classification formulations which are capable of handling interval data have immense importance from a pragmatic perspective. This paper presents a maximum-margin classification formulation which uses means and bounding hyper-rectangles (support) of the interval-valued training examples in order to build the decision function. As shown in the paper, the proposed classifier is robust to interval uncertainty and is also not overly-conservative.

The idea is to model interval-valued uncertainty using Chance-Constrained Programming (CCP). The main contribution of the paper is to approximate the CCP as a Second Order Cone Program (SOCP) using Bernstein schemes [4]. SOCPs are well studied convex optimization problems with efficient interior point solvers (e.g. `SeDuMi` [5]). The key advantage of the Bernstein scheme is that no assumptions on distributions regarding the underlying uncertainty are made and only partial information like support and mean of the uncertain examples is required. Geometric interpretation of the SOCP formulation reveals that the classifier views each example with interval uncertainty as a region of intersection of its bounding hyper-rectangle and an ellipsoid centered at its mean. Thus the proposed classifier is far less conservative than the methods which utilize the bounding hyper-rectangle information alone. Since a classifier's conservativeness directly affects the classification margin achieved, the proposed classifier is expected to generalize better. Methodology of classifying uncertain test examples is discussed and error measures for evaluating performance of interval data classifiers are presented. Experimental results show that the proposed classifier outperforms state-of-the-art when evaluated using any of the discussed error measures.

The paper is organized as follows: in section 2, the main contributions of the paper are presented. In section 3, experiments on real-world and synthetic data are presented. Section 4 summarizes the work and concludes the paper.

2 Robust Classifiers for Interval-Valued Uncertainty

This section presents the main contribution of the paper, a novel maximum-margin formulation for interval data in section 2.1. A discussion on geometric interpretation of the proposed formulation is presented in section 2.2. The section concludes with a discussion on error measures which evaluate the performance of a classifier on interval data.

2.1 Maximum-Margin Formulation using Bernstein Bounds

In this section, a maximum-margin classification formulation for interval data, which assumes the mean and the bounding hyper-rectangles are known for each example, is presented. It is also assumed that the features describing the data are independent. Let $\mathbf{X}_i = [X_{i1} \dots X_{in}]^\top$ be the random vector representing i^{th} training example (n denotes dimensionality of the data) and y_i denotes its label ($i = 1, \dots, m$). Let $\mathbf{a}_i = [a_{i1} \dots a_{in}]^\top$, $\mathbf{b}_i = [b_{i1} \dots b_{in}]^\top$ and $a_{ij} \leq X_{ij} \leq b_{ij}$,

so that $[\mathbf{a}_i, \mathbf{b}_i]$ represents the bounding hyper-rectangle of i^{th} example. Also let $\mathbf{E}[X]$ denote mean of the random variable X.

Consider the problem of constructing a maximum-margin classifier using the training example $\mathbf{X}_i$, which have interval-valued uncertainty. Let the discriminating hyperplane be denoted by $\mathbf{w}^\top \mathbf{x} - b = 0$. Then the constraints $y_i(\mathbf{w}^\top \mathbf{X}_i - b) \geq 1$ ensure that the training data is classified correctly. Slack variables, $\xi_i \geq 0$, can be introduced in order to handle outliers. Thus the classification constraints turn out to be $y_i(\mathbf{w}^\top \mathbf{X}_i - b) \geq 1 - \xi_i$. Since the constraints involve the random vector, X_i, they cannot be satisfied always. Hence, alternatively, one can ensure that the following chance-constraints are satisfied:

$$Prob\left(y_i(\mathbf{w}^\top \mathbf{X}_i - b) \leq 1 - \xi_i\right) \leq \epsilon \tag{1}$$

where $0 \leq \epsilon \leq 1$ is a small number denoting the upper bound on misclassification error made on an example and is a user-given parameter. Using these constraints the following maximum-margin formulation, similar in spirit to SVMs [6], can be written:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^m \xi_i \\ \text{s.t. } & Prob\left(y_i(\mathbf{w}^\top \mathbf{X}_i - b) \leq 1 - \xi_i\right) \leq \epsilon,\ \xi_i \geq 0,\ i = 1, \ldots, m \end{aligned} \tag{2}$$

In the following we will show that the CCP (2) can be approximated as an SOCP problem by using Bernstein bounds. To this end, the following theorem is presented, which specializes the Bernstein approximation schemes described in [4, 7]:

Theorem 1. *Consider the following notation* ($\forall\ i = 1, \ldots, m,\ j = 1, \ldots, n$)*:*

$$\begin{aligned} & l_{ij} = \frac{b_{ij} - a_{ij}}{2} && m_{ij} = \frac{b_{ij} + a_{ij}}{2} && \mu_{ij} = \frac{\mathbf{E}[X_{ij}] - m_{ij}}{l_{ij}} \\ & \mathbf{L}_i = diag\left([l_{i1} \ldots l_{in}]\right) && \mathbf{m}_i = [m_{i1} \ldots m_{in}]^\top && \mu_i = [\mu_{i1} \ldots \mu_{in}]^\top \\ & \mu_{ij}^{'} = \mathbf{E}[X_{ij}] && \mu_i^{'} = \left[\mu_{i1}^{'} \ldots \mu_{in}^{'}\right]^\top && \mathbf{\Sigma}_i = diag\left([\sigma(\mu_{i1}) \ldots \sigma(\mu_{in})]\right) \end{aligned} \tag{3}$$

where $\sigma(\mu_{ij})$ is given by:

$$\sigma(\mu_{ij}) = \min\left\{\sigma \geq 0 \;\middle|\; \frac{\sigma^2}{2}t^2 + \mu_{ij}t - \log\left(\cosh(t) + \mu_{ij}\sinh(t)\right) \geq 0,\ \forall\ t\ \in\ \mathbb{R}\right\} \tag{4}$$

The chance constraint (1), which represents the classification constraint for i^{th} example, is satisfied if the following cone constraint, in variables $\mathbf{w}, b, \xi_i, \mathbf{z}_i$, holds:

$$y_i(\mathbf{w}^\top \mu_i^{'} - b) + \mathbf{z}_i^\top \mu_i \geq 1 - \xi_i + \|\mathbf{z}_i\|_1 + \sqrt{2\log(1/\epsilon)}\,\|\mathbf{\Sigma}_i(y_i\mathbf{L}_i\mathbf{w} + \mathbf{z}_i)\|_2 \tag{5}$$

Proof. The chance-constraint (1) can be written as:

$$Prob\left(-y_i\mathbf{w}^\top\mathbf{X}_i + (1-\xi_i+y_ib) \geq 0\right) \leq \epsilon$$

Now, let variables $\mathbf{u}_i, \mathbf{v}_i$ be chosen such that:

$$\mathbf{u}_i + \mathbf{v}_i = -y_i\mathbf{w} \tag{6}$$

Since $\mathbf{a}_i \leq \mathbf{X}_i \leq \mathbf{b}_i$, we have that $\mathbf{v}_i^\top\mathbf{X}_i \leq \mathbf{v}_i^\top\mathbf{m}_i + \|\mathbf{L}_i\mathbf{v}_i\|_1$. Using this inequality, we have that the chance-constraint (1) is satisfied if:

$$Prob\left(\mathbf{u}_i^\top\mathbf{X}_i + u_{i0} \geq 0\right) \leq \epsilon \tag{7}$$

where $u_{i0} = 1-\xi_i+y_ib+\mathbf{v}_i^\top\mathbf{m}_i+\|\mathbf{L}_i\mathbf{v}_i\|_1$. Clearly, the advantage of introducing the variables $\mathbf{u}_i, \mathbf{v}_i$ (6) is to utilize the bounding hyper-rectangle information via the inequality $\mathbf{v}_i^\top\mathbf{X}_i \leq \mathbf{v}_i^\top\mathbf{m}_i + \|\mathbf{L}_i\mathbf{v}_i\|_1$ (also see lemma 2).

Using Markov inequality and independence of random variables $X_{ij}, j = 1,\ldots,n$, we have

$$Prob\left(\mathbf{u}_i^\top\mathbf{X}_i + u_{i0} \geq 0\right) \leq \exp\{\alpha u_{i0}\}\prod_j \mathbf{E}\left[\exp\{\alpha u_{ij}X_{ij}\}\right],\ \forall\ \alpha \geq 0 \tag{8}$$

The Key of modeling chance constraint (7) now depends on how one upper-bounds the moment generating functions $\mathbf{E}\left[\exp\{tX_{ij}\}\right],\ t \in \mathbb{R}$. To this end, we use the following lemma:

Lemma 1. *Consider the notation in (3). Then,*

$$\mathbf{E}\left[\exp\{tX_{ij}\}\right] \leq \exp\left\{\frac{\sigma(\mu_{ij})^2 l_{ij}^2}{2}t^2 + \mu_{ij}^{'}t\right\}\ \forall\ t\ \in\ \mathbb{R} \tag{9}$$

Proof. The fact that $\exp\{tX_{ij}\}$ is a convex function gives the following inequality: $\exp\{tX_{ij}\} \leq \frac{b_{ij}-X_{ij}}{b_{ij}-a_{ij}}\exp\{ta_{ij}\} + \frac{X_{ij}-a_{ij}}{b_{ij}-a_{ij}}\exp\{tb_{ij}\}$. Taking expectation on both sides and re-writing the resulting inequality in terms of m_{ij}, l_{ij} gives:

$$\mathbf{E}\left[\exp\{tX_{ij}\}\right] \leq \exp\left\{m_{ij}t + h_{ij}(l_{ij}t)\right\} \tag{10}$$

where $h_{ij}(\beta) \equiv \log\left(\cosh(\beta) + \mu_{ij}\sinh(\beta)\right)$. Note that, $h_{ij}(0) = 0, h_{ij}^{'}(0) = \mu_{ij}$ and $h_{ij}^{''}(\beta) \leq 1\ ,\ \forall\ \beta$. This gives the inequality $h_{ij}(\beta) \leq \frac{1}{2}\beta^2 + \mu_{ij}\beta\ ,\ \forall\ \beta$. In fact, using $\sigma(\mu_{ij})$ as defined in (4), we have the tighter inequality, $h_{ij}(\beta) \leq \frac{\sigma(\mu_{ij})^2}{2}\beta^2 + \mu_{ij}\beta\ ,\ \forall\ \beta$. Using this inequality in (10), and noting that $\mu_{ij}^{'} = l_{ij}\mu_{ij} + m_{ij}$, we obtain (9). This completes the proof of Lemma 1. □

Using Lemma 1, from (8) we obtain: $\log\left[Prob\left(\mathbf{u}_i^\top\mathbf{X}_i + u_{i0} \geq 0\right)\right] \leq \alpha(u_{i0} + \mathbf{u}_i^\top\mu_i^{'}) + \frac{\alpha^2}{2}\|\mathbf{L}_i\mathbf{\Sigma}_i\mathbf{u}_i\|_2^2\ ,\ \forall\ \alpha \geq 0$. Since this inequality holds for all values of α, if we ensure that for certain α the right-hand side of the inequality is $\leq \log(\epsilon)$, then we would satisfy the chance-constraint (7). Choosing $\alpha^* = -\frac{u_{i0}+\mathbf{u}_i^\top\mu_i^{'}}{\|\mathbf{L}_i\mathbf{\Sigma}_i\mathbf{u}_i\|_2^2}$, which is

the one minimizing right-hand side of the inequality, we see that (7) and in turn (1) are satisfied if:

$$u_{io} + \mathbf{u}_i^\top \mu_i^{'} + \sqrt{2\log(1/\epsilon)}\|\mathbf{L}_i\mathbf{\Sigma}_i\mathbf{u}_i\|_2 \leq 0 \tag{11}$$

Substituting the value of u_{i0}, eliminating the variable $\mathbf{u}_i$ from (6), (11) and introducing the variable $\mathbf{z}_i = \mathbf{L}_i\mathbf{v}_i$ gives (5). This completes the proof of the theorem. □

The values of $\sigma(\mu_{ij})$ (4) can be calculated numerically (refer Appendix A). Using theorem 1 and CCP (2), a maximum-margin SOCP formulation for interval data which ensures that the probability of misclassification is less than ϵ, can be written as follows:

$$\begin{array}{ll} \min\limits_{\mathbf{w},b,\mathbf{z}_i,\ \xi_i\geq 0} & \dfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum\limits_{i=1}^{m}\xi_i \\ \text{s.t.} & y_i(\mathbf{w}^\top\mu_i^{'} - b) + \mathbf{z}_i^\top\mu_i \geq 1 - \xi_i + \|\mathbf{z}_i\|_1 + \kappa\left\|\mathbf{\Sigma}_i(y_i\mathbf{L}_i\mathbf{w} + \mathbf{z}_i)\right\|_2 \end{array} \tag{12}$$

where $\kappa = \sqrt{2\log(1/\epsilon)}$ and $\mu_i^{'}, \mu_i, \mathbf{L}_i, \mathbf{\Sigma}_i$ are as given in (3). As mentioned earlier, C and ϵ are user-given parameters.

2.2 Geometric Interpretation of the Formulation

In this section, a geometrical interpretation for the proposed formulation (12) is presented. To this end, consider the following lemma:

Lemma 2. *Consider the notation in (3) and let* $\mathbf{S}_i = \mathbf{L}_i^2\mathbf{\Sigma}_i^2, \kappa = \sqrt{2\log(1/\epsilon)}$. *Suppose* $\mathcal{E}(\mu_i^{'}, \mathbf{S}_i, \kappa)$ *represents the ellipsoid* $\left\{\mathbf{x} = \mu_i^{'} + \kappa\mathbf{L}_i\mathbf{\Sigma}_i\mathbf{u} \ : \ \|\mathbf{u}\|_2 \leq 1\right\}$ *and* $\mathcal{R}(\mathbf{a}_i, \mathbf{b}_i)$ *represents the hyper-rectangle* $\{\mathbf{x} \ : \ \mathbf{a}_i \leq \mathbf{x} \leq \mathbf{b}_i\}$. *Consider the problem of correctly classifying points belonging to the intersection of* $\mathcal{E}(\mu_i^{'}, \mathbf{S}_i, \kappa)$ *and* $\mathcal{R}(\mathbf{a}_i, \mathbf{b}_i)$*:*

$$y_i(\mathbf{w}^\top\mathbf{x} - b) \geq 1 - \xi_i, \ \forall \ \mathbf{x} \ \in \ \mathcal{E}(\mu_i^{'}, \mathbf{S}_i, \kappa) \cap \mathcal{R}(\mathbf{a}_i, \mathbf{b}_i) \tag{13}$$

The continuum of constraints represented in (13) is satisfied if and only if the constraint (5) holds.

Proof. The constraint (13) hold if and only if:

$$1 - \xi_i + y_i b + \left(\max_{\mathbf{x} \ \in \ \mathcal{E}(\mu_i^{'}, \mathbf{S}_i, \kappa) \cap \mathcal{R}(\mathbf{a}_i, \mathbf{b}_i)} -y_i\mathbf{w}^\top\mathbf{x}\right) \leq 0$$

Note that, $\max_{\mathbf{x} \ \in \ \mathcal{E}(\mu_i^{'}, \mathbf{S}_i, \kappa) \cap \mathcal{R}(\mathbf{a}_i, \mathbf{b}_i)} \left(-y_i\mathbf{w}^\top\mathbf{x}\right)$ is the support function of the set $\mathcal{E}(\mu_i^{'}, \mathbf{S}_i, \kappa) \cap \mathcal{R}(\mathbf{a}_i, \mathbf{b}_i)$ (denoted by $I_{\mathcal{E}(\mu_i^{'}, \mathbf{S}_i, \kappa) \cap \mathcal{R}(\mathbf{a}_i, \mathbf{b}_i)}(-y_i\mathbf{w})$). Since support

function of intersection of two sets is the infimal convolution of support functions of the individual sets (see section 16, [8]), we have that $I_{\mathcal{E}(\mu_i^{'},\mathbf{S}_i,\kappa)\cap\mathcal{R}(\mathbf{a}_i,\mathbf{b}_i)}(-y_i\mathbf{w}) = \inf\left\{I_{\mathcal{E}(\mu_i^{'},\mathbf{S}_i,\kappa)}(\mathbf{u}_i) + I_{\mathcal{R}(\mathbf{a}_i,\mathbf{b}_i)}(\mathbf{v}_i) \mid \mathbf{u}_i + \mathbf{v}_i = -y_i\mathbf{w}\right\}$. Thus we have:

$$
\begin{aligned}
(13) \Leftrightarrow 1 - \xi_i + y_i b + \inf\left\{I_{\mathcal{E}(\mu_i^{'},\mathbf{S}_i,\kappa)}(\mathbf{u}_i) + I_{\mathcal{R}(\mathbf{a}_i,\mathbf{b}_i)}(\mathbf{v}_i) \mid \mathbf{u}_i + \mathbf{v}_i = -y_i\mathbf{w}\right\} \leq 0 \\
\Leftrightarrow \exists\ \mathbf{u}_i, \mathbf{v}_i \ \ni\ \mathbf{u}_i + \mathbf{v}_i = -y_i\mathbf{w}, \\
1 - \xi_i + y_i b + \left\{I_{\mathcal{E}(\mu_i^{'},\mathbf{S}_i,\kappa)}(\mathbf{u}_i) + I_{\mathcal{R}(\mathbf{a}_i,\mathbf{b}_i)}(\mathbf{v}_i)\right\} \leq 0 \qquad (14)
\end{aligned}
$$

Now it is easy to see that $I_{\mathcal{E}(\mu_i^{'},\mathbf{S}_i,\kappa)}(\mathbf{u}_i) = \mathbf{u}_i^\top\mu_i^{'} + \kappa\|\mathbf{L}_i\mathbf{\Sigma}_i\mathbf{u}_i\|_2$ and $I_{\mathcal{R}(\mathbf{a}_i,\mathbf{b}_i)}(\mathbf{v}_i) = \mathbf{v}_i^\top\mathbf{m}_i + \|\mathbf{L}_i\mathbf{v}_i\|_1$. Substituting these values in (14) and noting that $\mu_i^{'} = \mathbf{L}_i\mu_i + \mathbf{m}_i$, gives (6, 11). Since the constraints (6, 11) are equivalent to (5) (see proof of theorem 1), we have that (13) $\Leftrightarrow$ (5). This completes the proof. □

The above lemma shows that the proposed classifier (12) views each interval data example as the intersection of the bounding hyper-rectangle and an ellipsoid centered at its mean with semi-axis lengths proportional to $l_{ij}, \sigma(\mu_{ij})$. In this way the proposed formulation takes into account both the mean and bounding hyper-rectangle informations. Note that, lemma 2 theoretically proves that the proposed classifier is always less conservative (pessimistic) than classifiers which use the bounding hyper-rectangle information alone [3] (this is because $\mathcal{E}(\mu_i^{'},\mathbf{S}_i,\kappa)\cap\mathcal{R}(\mathbf{a}_i,\mathbf{b}_i) \subset \mathcal{R}(\mathbf{a}_i,\mathbf{b}_i)$). In fact, it is easy to see that classifiers which utilize the bounding hyper-rectangle information alone and classifiers which utilize the mean information alone are special cases of the proposed formulation (with $\epsilon = 0$ and $\epsilon = 1$ respectively).

It is interesting to note the effect of $\sigma(\mu_{ij})$ (4) on the proposed classifier. As mentioned earlier, the semi-axis lengths of the uncertainty ellipsoid are proportional to $\sigma(\mu_{ij})$. Table 3 shows that as μ increases from 0 to 1, $\sigma(\mu)$ decreases from 1 to 0. In other words, as the mean of example shifts from center to a corner of the bounding hyper-rectangle, the size of the uncertainty ellipsoid decreases. This is very intuitive because, in one extreme case where mean lies at a corner, the datapoint is deterministic and in the other extreme case, where mean is at center of the hyper-rectangle, the uncertainty of the datapoint is maximum. This phenomenon is also illustrated in figure 1, where the bounding hyper-rectangle and the uncertainty region at various positions of the mean are shown. It can be seen that as the mean moves towards a corner, not only does the uncertainty region move but also its size decreases. However, a classifier which uses does not employ the mean information [3], always views the example as the whole hyper-rectangle. Thus the proposed classifier is robust to interval-valued uncertainty, as well as not overly-conservative.

2.3 Classification with Uncertain Examples

This section presents the methodology for labeling interval-valued test examples and discusses error measures for evaluating the performance of interval data classifiers. Depending on the form in which the examples are available, the labeling

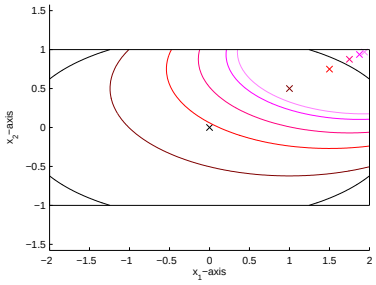

Fig. 1. Figure showing bounding hyper-rectangle and uncertainty sets for different positions of mean. Mean and boundary of uncertainty set marked with same color.

methodologies summarized in table 1 can be employed. Here, y_i^{pr} denotes the predicted label for test example $\mathbf{X}_i$ (also refer (3) for notation). Once a test example is labeled using the appropriate methodology, the overall misclassification error for the given test dataset can be calculated as the percentage of examples in which y_i^{pr} and y_i do not agree:

$$\textbf{NomErr} = \frac{\sum_i 1_{y_i^{pr} \neq y_i}}{\#\ test\ examples} \times 100 \tag{15}$$

Note that, the proposed formulation can be solved when the training examples are in any of the 3 forms shown in table 1. In case of form 2, the support and mean information are readily available and in case of form 3 these partial information can be easily estimated from the replicates. In case of form 1, since no mean information is available the proposed formulation can be solved using $\epsilon = 0$, which as discussed in section 2.2 is the maximum-margin classifier built using support information alone.

Table 1. Table summarizing ways of representing interval-valued uncertainty in examples and corresponding label prediction methodologies.

S.No.	Form of examples	Labeling Methodology
1	$\mathcal{R}(\mathbf{a}_i, \mathbf{b}_i)$ are given	$y_i^{pr} \equiv \text{sign}(\mathbf{w}^\top \mathbf{m}_i - b)$
2	$\mathcal{R}(\mathbf{a}_i, \mathbf{b}_i), \mu_i^{'}$ are given	$y_i^{pr} \equiv \text{sign}(\mathbf{w}^\top \mu_i^{'} - b)$
3	Set of replicates $\mathbf{r}_{i1}, \mathbf{r}_{i2}, \ldots$	y_i^{pr} is label of majority of $\mathbf{r}_{ij}$. Label of $\mathbf{r}_{ij}$ is $\text{sign}(\mathbf{w}^\top \mathbf{r}_{ij} - b)$

Based on the discussion in section 2.2, another interesting error measure can be derived. Given an uncertain test example X_i with label y_i, one can calculate the value of $\epsilon = \epsilon_{opt} = \exp\left\{-\frac{(\mathbf{w}^\top \mu_i^{'} - b)^2}{2(\mathbf{w}^\top \mathbf{L}_i^2 \mathbf{\Sigma}_i^2 \mathbf{w})}\right\}$ for which the uncertainty ellipsoid $\mathcal{E}(\mu_i^{'}, \mathbf{S}_i, \kappa)$ touches the discriminating hyperplane, $\mathbf{w}^\top \mathbf{x} - b = 0$. Additionally,

if true label, y_i, of the example is same as the predicted label (y_i^{pr}), then the proof of theorem 1 shows that the true probability of misclassification of the test example is less than or equal to ϵ_{opt}. This leads to the following definition of error on each test example:

$$\mathbf{OptErr}_i = \begin{cases} 1 & \text{if } y_i \neq y_i^{pr} \\ \epsilon_{opt} & \text{if } y_i = y_i^{pr} \text{ and } \exists\ \mathbf{x} \in \mathcal{R}(\mathbf{a}_i, \mathbf{b}_i) \ni y_i(\mathbf{w}^\top \mathbf{x} - b) < 0 \\ 0 & y_i(\mathbf{w}^\top \mathbf{x} - b) \geq 0\ \forall\ \mathbf{x} \in \mathcal{R}(\mathbf{a}_i, \mathbf{b}_i) \end{cases} \tag{16}$$

The overall error, **OptErr**, can be calculated as percentage of $\mathbf{OptErr}_i$ over all test examples:

$$\mathbf{OptErr} = \frac{\sum_i \mathbf{OptErr}_i}{\#\ test\ examples} \times 100 \tag{17}$$

Note that, both **NomErr** and **OptErr** can be estimated for any hyperplane classifier and are not specific to the proposed classifier. Experimental results show that the proposed classifier achieves lower **NomErr** and **OptErr** when compared to existing interval data classification methods.

3 Numerical Experiments

In this section, experimental results which compare performance of the proposed interval data classifier (12) (denoted by **IC-MBH**) and the maximum-margin classifier which utilizes bounding hyper-rectangle information alone [3] (denoted by **IC-BH**):

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\textstyle\sum_{i=1}^m \xi_i \\ \text{s.t. } & y_i(\mathbf{w}^\top \mathbf{m}_i - b) \geq 1 - \xi_i + \|\mathbf{L}_i\mathbf{w}\|_1,\ \xi_i \geq 0,\ i = 1, \ldots, m \end{aligned} \tag{18}$$

are presented. Note that the only difference between (18) and the one proposed in [3] is minimization of $\|\mathbf{w}\|_2$ in the objective rather than minimization of $\|\mathbf{w}\|_1$, which implies maximum-margin classification rather than sparse classification. We have done this in order to achieve a fair comparison of the methodologies. Traditional classifiers like SVMs cannot handle interval-valued data. However, in cases where the means of the uncertain examples are known or in cases where uncertainty is represented using replicates (e.g. form 2 and 3 in table 1 respectively), SVMs can be trained by considering mean of each example as a training datapoint or by considering each replicate as a training datapoint. Henceforth, let these classifiers be denoted by **IC-M** and **IC-R** respectively. Hence, wherever applicable, we compare the performance of the proposed classifier with SVM based methods also.

Experiments were performed on synthetic datasets and two real-world datasets: micro-array data[4] [1] and Wisconsin Diagnostic Breast Cancer (WDBC) dataset[5].

[4] Available at `http://www.ncbi.nlm.nih.gov/geo/` with accession number GSE2187.
[5] Available at `http://mlearn.ics.uci.edu/MLSummary.html`

Synthetic datasets were generated using the following methodology: a) nominal (true) examples were generated using Gaussian mixture models b) uncertainty was introduced into each nominal point using standard finite-supported distributions (whose parameters are chosen randomly) c) replicates for each nominal example were produced by sampling the chosen noise distribution. The synthetic datasets are named using dimension of the dataset and are subscripted with the distribution used for generating replicates (e.g. synthetic data of dimensionality n with Uniform, truncated skew-Normal and truncated Beta noise distributions are denoted by $\mathbf{n_U}$, $\mathbf{n_{SN}}$ and $\mathbf{n}_\beta$ respectively). In each case, a synthetic test dataset was also generated independently.

The micro-array dataset defines a 4 category classification task, namely that of identifying four kinds of drugs: Azoles ($\mathcal{A}$), Fibrates ($\mathcal{F}$), Statins ($\mathcal{S}$) and Toxicants ($\mathcal{T}$). Instead of handling a multi-class problem we have defined six binary classification tasks using "one versus one" scheme. As a preprocessing step we have reduced the dimension of the problem to 200 by feature selection using Fisher score. In case of both synthetic and micro-array data, the means and bounding hyper-rectangles were estimated from the replicates provided for each training example.

The task of WDBC is to classify "benign" and "malignant" tumours based on 10 features computed from tumour cell nuclei. However, since the measurements are not the same over all tumour cells, the mean, standard-error and maximum values of the 10 features are provided. From this information the bounding hyper-rectangles and means for each training example are estimated.

In section 3.1, we compare classification margins ($2/\|\mathbf{w}\|_2$) achieved by **IC-BH** and **IC-MBH**, which represent state-of-the-art and the proposed interval data classifier respectively. The key results of the paper are presented in section 3.2. These results compare the **NomErr** (15) and **OptErr** (17) obtained with various classifiers.

3.1 Comparison of Margin

In this section, the margins ($2/\|\mathbf{w}\|_2$) achieved by **IC-BH** and **IC-MBH** at a fixed value of the C parameter are compared. Figure 2 summarizes the results. Note that, at all values of ϵ, the classification margin with **IC-MBH** is higher than that with **IC-BH**. Also, as the value of ϵ or dimensionality of the data increases, difference in the margins achieved by **IC-BH** and **IC-MBH** also increases. The explanation for this is clear from the geometric interpretation presented in section 2.2. According to Structural Risk minimization principle of Vapnik [6], higher margin implies better generalization. Hence the proposed classifier is expected to achieve good generalization for interval data. As a baseline for comparison, the margin achieved by the SVM trained using means of the examples, **IC-M**, is also shown in the figure. Since **IC-M** does not take into account the interval uncertainty and assumes the mean to be the "true" training example, it always achieves higher margin than **IC-BH, IC-MBH**. The trend shown in figure 2 remained the same for higher dimensions and with different noise distributions ($\mathbf{n_{SN}}, \mathbf{n}_\beta$).

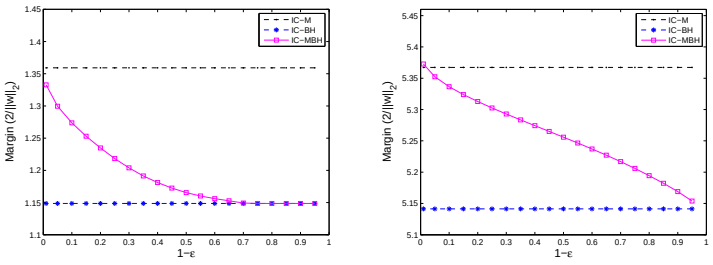

Fig. 2. Figure comparing margins achieved by **IC-M, IC-BH** and **IC-MBH** at various ϵ values ($\mathbf{2_U}$ on left and $\mathbf{10_U}$ on right).

3.2 Comparison of Generalization Error

This section presents results which compare the performance of **IC-M, IC-R, IC-BH** and **IC-MBH** when evaluated using the error measures **NomErr** (15) and **OptErr** (17). Experiments were done on the synthetic and real-world datasets described in section 3. In all cases, the hyper-parameters (C and/or ϵ) for each classifier were tuned using a 3-fold cross-validation procedure. The results are summarized in table 2. In case of synthetic datasets, the reported values represent the mean testset error achieved with the tuned hyper-parameters when trained with 10 different training sets each generated from the same synthetic data template. In case of the real-world datasets, the values represent cross-validation error with tuned hyper-parameters averaged over three cross-validation experiments. Hence the error values reported in the table represent a good estimate of the generalization error of the respective classifiers. Clearly, **NomErr** and **OptErr** are least for **IC-MBH**; confirming that **IC-MBH** achieves good generalization for interval data. Moreover, in case of many datasets, the proposed classifier outperforms the existing classifiers in terms of the **OptErr** error measure.

4 Conclusions

This paper presents a novel maximum-margin classifier which achieves good generalization on data having interval-valued uncertainty. The key idea was to employ chance-constraints in order to handle uncertainty. The main contribution was to derive a tractable SOCP formulation, which is a safe approximation of the resulting CCP, using Bernstein schemes. The formulation ensures that the probability of misclassification on interval data is less than a user specified upper-bound (ϵ). Also, the geometric interpretation shows that the classifier views each training example as the region of intersection of its bounding hyper-rectangle and an ellipsoid centered at its mean.

The proposed classifier is robust to interval-valued uncertainty and is also not overly-conservative. As shown in the paper, this amounts to achieving higher

Table 2. Table comparing **NomErr** and **OptErr** obtained with **IC-M, IC-R, IC-BH** and **IC-MBH**.

Dataset	IC-M		IC-R		IC-BH		IC-MBH	
	NomErr	**OptErr**	**NomErr**	**OptErr**	**NomErr**	**OptErr**	**NomErr**	**OptErr**
$\mathbf{10_U}$	32.07	59.90	44.80	65.70	51.05	53.62	**20.36**	**52.68**
$\mathbf{10}_\beta$	46.46	54.78	48.02	53.52	46.67	49.50	**46.18**	**49.38**
$\mathcal{A}$ vs. $\mathcal{F}$	00.75	46.47	00.08	46.41	55.29	58.14	**00.07**	**39.68**
$\mathcal{A}$ vs. $\mathcal{S}$	09.02	64.64	08.65	68.56	61.69	61.69	**06.10**	**39.63**
$\mathcal{A}$ vs. $\mathcal{T}$	12.92	73.88	**07.92**	81.16	58.33	58.33	11.25	**40.84**
$\mathcal{F}$ vs. $\mathcal{S}$	01.03	34.86	00.95	38.73	28.21	49.25	**00.05**	**27.40**
$\mathcal{F}$ vs. $\mathcal{T}$	06.55	55.02	05.81	58.25	51.19	60.04	**05.28**	**35.07**
$\mathcal{S}$ vs. $\mathcal{T}$	10.95	64.71	**05.00**	70.76	69.29	69.29	**05.00**	**30.71**
WDBC	55.67	**37.26**	×	×	**37.26**	45.82	47.04	45.84

classification margins and in turn better generalization than methods which employ the bounding hyper-rectangle information alone. As the results showed, the average error with the proposed classifier, in case of many synthetic and real-world datasets, is less than half of that with the existing methods.

The Bernstein relaxations schemes presented in this paper not only aid in approximating the original CCP as a convex program, but also open avenues for efficient approximations of other CCP-based learning formulations (e.g. [9] and its variants). By employing rich partial information, the Bernstein schemes lead to less conservative relaxations. Hence exploitation of the Bernstein schemes in the context of learning is a good direction for research.

5 Acknowledgment

SB and CB is supported by a Yahoo! faculty grant.

A Computation of $\boldsymbol{\sigma(\mu)}$

In this section, we present details of the numerical procedure for computing $\sigma(\mu)$. Consider the following claim:

Claim. Let $\sigma(\mu)$ be as defined in (4). Then, $\sqrt{1-\mu^2} \leq \sigma(\mu) \leq 1$.

Proof. Recalling the definition of $\sigma(\mu)$, we have:

$$\sigma(\mu) = \min \{\sigma \geq 0 \mid f(t;\sigma,\mu) \geq 0,\ \forall\ t\ \in\ \mathbb{R}\}$$

where $f(t;\sigma,\mu) \equiv \frac{\sigma^2}{2}t^2 + \mu t - \log\left(\cosh(t) + \mu \sinh(t)\right)$. Let $f^{'}(t;\sigma,\mu) = g_1(t) - g_2(t)$ where $g_1(t) \equiv \sigma^2 t + \mu$ and $g_2(t) \equiv \frac{\sinh(t)+\mu\cosh(t)}{\cosh(t)+\mu\sinh(t)}$. Now, if $g_1^{'}(0) < g_2^{'}(0)$, then there exists a neighbourhood around $t = 0$ where $f^{'}(t) < 0$ (since $f^{'}(0) = 0$). Also in this neighbourhood $f(t) < 0$ because $f(0) = 0$. Thus $g_1^{'}(0) \geq g_2^{'}(0)$ is a necessary condition for $f \geq 0$. In other words, $\sigma(\mu) \geq \sqrt{1-\mu^2}$. Also, from proof of lemma 2 we have that $\sigma(\mu) \leq 1$. This completes the proof. □

Note that, the function f strictly increases with the value of σ and by claim A we have that $\sqrt{1-\mu^2} \leq \sigma(\mu) \leq 1$. Thus one can have a simple binary search algorithm for computing σ. The algorithm starts with $\sigma_0^l \equiv \sqrt{1-\mu^2}$ and $\sigma_0^u \equiv 1$. At every iteration, $i \geq 1$, $\sigma_i \equiv \frac{\sigma_{i-1}^l + \sigma_{i-1}^u}{2}$ and it is checked whether $f_i^{min} \equiv (\min_t f(t; \sigma_i, \mu)) \geq 0$. If $f_i^{min} \geq 0$, then $\sigma_i^u \equiv \sigma_i$, else $\sigma_i^l \equiv \sigma_i$. This is repeated until a relevant stopping criteria is met. Also, as the proof of claim A suggests, for fixed values of σ, μ, the function f has only one minimum wrt. t (this is because $g_2(t)$ is concave above $t = t^*$ and convex below $t = t^*$). Hence checking whether $f_i^{min} \geq 0$ for a fixed value σ_i is also easy. The values of σ as a function of $\mu \in [0, 1]$ are shown in table 3. Since the function $\sigma(\mu)$ is symmetric wrt. μ, we have $\sigma(\mu) = \sigma(-\mu)$.

Table 3. Table showing values of σ as a function of $\mu \in [0, 1]$ at 20 equal increments.

1.0000	0.9995	0.9979	0.9958	0.9914	0.9876	0.9827	0.9745	0.9627	0.9560
0.9481	0.9356	0.9176	0.8927	0.8686	0.8538	0.8279	0.7812	0.6986	0.0000

References

1. G. Natsoulis, L. E. Ghaoui, G. R. G. Lanckriet, A. M. Tolley, F. Leroy, S. Dunlea, B. P. Eynon, C. I. Pearson, S. Tugendreich, and K. Jarnagin. Classification of a Large Microarray Data Set: Algorithm Comparison and Analysis of Drug Signatures. *Genome Research*, 15:724–736, 2005.
2. F. C. D. Silva, F. de A. T. de Carvalho, R. M. C. R. de Souza, and J. Q. Silva. A Modal Symbolic Classifier for Interval Data. In *ICONIP*, pages 50–59, 2006.
3. L. E. Ghaoui, G. R. G. Lanckriet, and G. Natsoulis. Robust Classification with Interval Data. Technical Report UCB/CSD-03-1279, Computer Science Division, University of California, Berkeley, 2003.
4. A. Nemirovski and A. Shapiro. Convex Approximations of Chance Constrained Programs. *SIAM Journal of Optimization*, 17(4):969–996, 2006.
5. J. F. Sturm. Using SeDuMi 1.02, A MATLAB Toolbox for Optimization over Symmetric Cones. *Optimization Methods and Software*, 11–12:625–653, 1999.
6. V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.
7. A. Ben-Tal and A. Nemirovski. Selected Topics in Robust Convex Optimization. *Mathematical Programming*, 112(1), 2007.
8. R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.
9. G. R. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A Robust Minimax Approach to Classification. *JMLR*, 3:555–582, 2003.