

Iterative Constraints in Support Vector Classification with Uncertain Information

Jianqiang Yang and Steve Gunn

School of Electronics and Computer Science, University of Southampton
Building 1, Highfield Campus, Southampton, SO17 1BJ, UK
{jy03r, srg}@ecs.soton.ac.uk

Abstract. This paper proposes a new iterative approach of input uncertainty classification, which incorporates input uncertainty information and exploits adaptive constraints extracted from uncertain inputs statistically and geometrically to extend the traditional support vector classification (SVC). Kernel functions can be implemented by a novel kernelized formulation to generalize this proposed technique to non-linear models and the resulting optimization problem is a second order cone program (SOCP) with a unique solution. Results demonstrate how this technique has an improved performance and is more robust than the traditional algorithms when uncertain information is available.

Key words: SVC, iterative constraints, uncertain, kernel functions

1 Introduction

Uncertain information associated with data is often ignored in traditional machine learning algorithms. Many approaches attempt to model any uncertainty in the form of additive noise on the target, which can be effective for simple models. However, for more complex non-linear models and where a richer description of anisotropic uncertainty in the input space is available, these approaches can suffer. For instance, the traditional support vector classification (SVC) can only accommodate isotropic uncertainty information in the input space.

Recent advances in machine learning methods have seen significant contribution from kernel-based approaches. These have many advantages, including strong theory and convex optimization formulation. Support vector machines (SVMs) are one approach that have been extended to incorporate uncertain data. Many other algorithms are also focusing on input uncertainty classification by implementing their own constraints. The rest of the paper explores an extension to SVC to provide a more robust algorithm, which enables uncertain information in the inputs to be incorporated iteratively into the constraints. The resulting algorithm is formulated as a second order cone programming (SOCP) optimization problem with adaptive constraints driven by the uncertainties.

The paper is organized as follows: section 2 presents the input uncertainty formulation for the classification task. In Sect.3, the dual problem is derived by introducing noise-specific covariance information as additional constraints

and the approach is extended to non-linear classification by a novel kernelized formulation. It is then shown how these geometric and statistical characteristics can be extended to generate two more efficient iterative algorithms in Sect. 4. In Sect. 5, some kernel functions are introduced along with the experimental results of these new approaches to compare with traditional algorithms.

2 Input Uncertainty Classification

Definition 1. Let $\mathcal{D} = \{z_i, y_i\}, i = 1, \dots, l$ denote the observed inputs, where $y_i \in \{-1, +1\}$, $z_i \in \mathbb{R}^n$ and $z_i \sim \mathcal{N}(x_i, \mathbf{M}_i)$, in which \mathcal{N} is a Gaussian distribution with mean $x_i \in \mathbb{R}^n$ and covariance $\mathbf{M}_i \in \mathbb{R}^{n \times n}$.

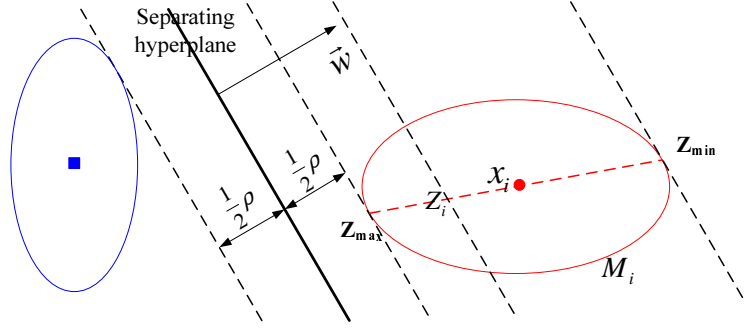


Fig. 1. The classification of Gaussian uncertainties in the input space ($n = 2$).

The input uncertainties in Definition 1 are shown in Fig. 1 [10], in which the ellipsoids represent the Gaussian distributions of the input uncertainties, ρ represents the margin between the closest edges of the ellipsoids to the optimal hyperplane, z_{\max} and z_{\min} represent those points, at which the hyperplanes parallel to the optimal hyperplane are tangent to the edges of the ellipsoids.

2.1 Geometric Interpretation

Let $\mathcal{E}(\mathbf{A}, \mathbf{a}) \subseteq \mathbb{R}^n$ denote an ellipsoid, $\mathbf{A} \in \mathbb{R}_+^{n \times n}$, $\mathbf{a} \in \mathbb{R}^n$ and $\mathcal{E}(\mathbf{A}, \mathbf{a}) := \{x \in \mathbb{R}^n \mid (x - \mathbf{a})^T \mathbf{A}^{-1} (x - \mathbf{a}) \leq 1\}$. Setting $\mathbf{Q}_i = \mathbf{M}_i^{1/2}$, according to Definition 1 and the theorem [4], which shows that every ellipsoid is the image of the unit ball around zero under a bijective affine transformation, we have

$$\begin{aligned} \max_w \{w^T z_i \mid z_i \in \mathcal{E}(\mathbf{M}_i, x_i)\} &= \max_w \{w^T \mathbf{Q}_i \mathbf{Q}_i^{-1} z_i \mid \mathbf{Q}_i^{-1} z_i \in \mathbf{Q}_i^{-1} \mathcal{E}(\mathbf{M}_i, x_i)\} \\ &= w^T \frac{1}{\sqrt{w^T \mathbf{M}_i w}} \mathbf{M}_i w + w^T x_i, \end{aligned} \quad (1)$$

where the optimal result is $\mathbf{z}_{\max} = \mathbf{x}_i + \frac{1}{\sqrt{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}} \mathbf{M}_i \mathbf{w}$, $\mathbf{z}_{\min} = \mathbf{x}_i - \frac{1}{\sqrt{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}} \mathbf{M}_i \mathbf{w}$ and consequently, $\mathbf{z}_i = \mathbf{x}_i + r \frac{1}{\sqrt{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}} \mathbf{M}_i \mathbf{w}$, $-1 \leq r \leq 1$. Figure 1 shows how \mathbf{z}_i follows its Gaussian distribution represented by an ellipsoid \mathbf{M}_i as r varies. The next section introduces a theorem on probabilistic linear inequalities, which enables a formulation of the extended uncertainty information.

2.2 Minimax Probability Machine

The minimax probability machine (MPM)[7] is a recent method introduced for pattern classification. MPM chooses a discriminative approach to minimize the misclassification probability of the future inputs without the prior knowledge of the distributions of inputs. MPM uses a theorem [1], which provides the stronger upper optimal bounds in probability than the result in Chebyshev's inequality, to derive an approach, transforming the probability inequality $\inf_{\mathbf{u} \sim (\bar{\mathbf{u}}, \Sigma_{\mathbf{u}})} \Pr\{\mathbf{a}^T \mathbf{u} \leq h\} \geq \alpha$ to the following expression without the probability in the inequality.

$$h - \mathbf{a}^T \bar{\mathbf{u}} \geq \kappa(\alpha) \sqrt{\mathbf{a}^T \Sigma_{\mathbf{u}} \mathbf{a}} \quad \text{where } \kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}. \quad (2)$$

where $\mathbf{a}^T \mathbf{u} \leq h$ represents a hyperplane, $\mathbf{a}, \mathbf{u} \in \mathbb{R}^n$, $h \in \mathbb{R}$, α is the inferior probability of the correctly classified inputs, $\bar{\mathbf{u}} \in \mathbb{R}^n$ is the mean and $\Sigma_{\mathbf{u}} \in \mathbb{R}^{n \times n}$ is the covariance of the inputs of a class.

2.3 Statistical Approach

First, we exploit theorem [1] to develop a formulation in which the probability of misclassification is minimized under this extended uncertainty description. We can use (2) to extend SVC to incorporate the uncertainty in Definition 1,

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \geq \kappa(\alpha) \sqrt{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}, \quad (3)$$

where $\kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$. Furthermore, we can consider a Gaussian model for the uncertainties on the inputs, we can transform $\inf_{\mathbf{u} \sim (\bar{\mathbf{u}}, \Sigma_{\mathbf{u}})} \Pr\{\mathbf{a}^T \mathbf{u} \leq h\}$ to

$$\inf_{\mathbf{z}_i \sim \mathcal{N}(\mathbf{x}_i, \mathbf{M}_i)} \Pr\{-y_i \mathbf{w}^T \mathbf{z}_i \leq y_i b - 1 + \xi_i\} = \Phi \left(\frac{y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i}{\sqrt{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}} \right) \geq \alpha, \quad (4)$$

where $\Phi(v) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^v \exp(-s^2/2) ds$ is the cumulative distribution function for a standard normal Gaussian distribution. Since $\Phi(v)$ is monotone increasing, we can write (4) as:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \geq \Phi^{-1}(\alpha) \sqrt{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}. \quad (5)$$

We can generate SVC constraints independent of the distributions of the uncertain inputs by combining (3) and (5),

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \geq r \sqrt{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}, \quad (6)$$

where $r \in \mathbb{R}$ is the probability confidence. Although the distribution is assumed to be Gaussian in this paper, (6) provides us with a way to exploit other distributions of the uncertain inputs when this information is available.

2.4 Missing Features

In some cases, uncertainties may be partially unknown [8]. Consider the Gaussian distribution $\mathcal{N}(\mathbf{x}_i, \mathbf{M}_i)$ introduced in Definition 1 where some features of \mathbf{x}_i are missing, and as the result, only part of the covariance matrix \mathbf{M}_i is known. We then can extend the Gaussian approximation from [3] to estimate the unknown components. Let \mathbf{x}_{ik} and \mathbf{x}_{im} denote the known features and the missing features of \mathbf{x}_i , and $\mathbf{x}_i = [\mathbf{x}_{ik}^T, \mathbf{x}_{im}^T]^T$. Introducing function $f(\mathbf{z}) = \mathbf{z}$, $\mathbf{z} \in \mathbb{R}^j$, $j = 1, \dots, n$, we then obtain

$$\begin{aligned} f(\mathbf{x}_i) &\sim \mathcal{N}(\mu_{\mathbf{x}_i}, \Sigma_{\mathbf{x}_i}) = \mathcal{N}(\mathbf{x}_i, \mathbf{M}_i) \\ \text{Cov}(f(\mathbf{x}_p), f(\mathbf{x}_q)) &= \text{Cov}(\mathbf{x}_p, \mathbf{x}_q) , \end{aligned} \quad (7)$$

where $\mathbf{x}_p, \mathbf{x}_q \in \mathbb{R}^m$, $m \leq n$ are parts of \mathbf{x}_i , and the covariance matrix $\text{Cov}(\mathbf{x}_p, \mathbf{x}_q)$ is part of \mathbf{M}_i . Then the distribution $p(\mathbf{x}_{ik}, \mathbf{x}_{im} | \mathbf{x}_{ik})$ is a Gaussian distribution with mean $[\mathbf{x}_{ik}^T, \mathbf{x}_{im}^T]^T$ and covariance matrix \mathbf{M}_i , which is shown as follows:

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k \end{bmatrix} , \quad (8)$$

where \mathbf{K} , \mathbf{k} and k are the covariance matrices of \mathbf{x}_{ik} , \mathbf{x}_{ik} and \mathbf{x}_{im} , and \mathbf{x}_{im} respectively. The predictive distribution of \mathbf{x}_{im} is

$$\mathbf{x}_{im} | \mathbf{x}_{ik} \sim \mathcal{N}(\mu(\mathbf{x}_{ik}), \Sigma(\mathbf{x}_{ik})) , \quad (9)$$

where $\mu(\mathbf{x}_{ik})$ and $\Sigma(\mathbf{x}_{ik})$ are obtained by:

$$\begin{aligned} \mu(\mathbf{x}_{ik}) &= \mathbf{x}_{im} + \mathbf{k}^T \mathbf{K}^{-1} (\mathbf{x}_k - \mathbf{x}_{ik}) \\ \Sigma(\mathbf{x}_{ik}) &= k - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} . \end{aligned} \quad (10)$$

An iterated algorithm to approximate the missing features is shown as follows:

Algorithm 1. Missing Features Approximation

When \mathbf{x}_{ik} converge to \mathbf{x}_k , $\mu(\mathbf{x}_{ik})$ and $\Sigma(\mathbf{x}_{ik})$ are what we want:

1. Initialize $\mathbf{x}_i = [\mathbf{x}_{iko}^T, \mathbf{x}_{imo}^T]^T$ and \mathbf{M}_i ;
2. Let $\mathbf{x}_k = \mathbf{x}_{ik}$, $\mathbf{x}_{ik} = \mathbf{x}_{iko}$ and $\mathbf{x}_{im} = \mathbf{x}_{imo}$, obtain \mathbf{K} , \mathbf{k} , k from \mathbf{M}_i and compute (10);
3. Recompute and collect the new value of \mathbf{x}_i and \mathbf{M}_i by using the completed data, allocate the new value to \mathbf{x}_{iko} and \mathbf{x}_{imo} , then return to step 2;

3 Uncertainty Support Vector Classification

In this section we derive the primal and dual formulations of the input uncertainty classification which is named uncertainty support vector classification (USVC).

3.1 Linear Case

The primal problem of USVC is obtained by introducing the constraints from (6) as:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi_i} \quad & \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^l \xi_i \\ \text{s.t.} \quad & r \|\mathbf{M}_i^{1/2} \mathbf{w}\| \leq y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \\ & r \geq 0 \quad \xi_i \geq 0 \quad i = 1, \dots, l \end{aligned} \quad (11)$$

where r needs to be set in advance in the optimization. USVC is a second order cone program (SOCP). Following the Lagrangian method, we have $\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i + \sum_{i=1}^l (\mathbf{M}_i^{1/2})^T \beta_i$, in which the dual variables, $\beta_i \in \mathbb{R}^n$, $i = 1, \dots, l$ control the influence of the covariance matrices describing the distributions of the uncertain inputs, while the dual variables, $\alpha_i \in \mathbb{R}$, $i = 1, \dots, l$, behave in a similar manner to the SVC. When $r = 0$, meaning that the probability of the examples being correctly classified is set to 0.5 in the classification, or $\mathbf{M}_i = \mathbf{0}$, $i = 1, \dots, l$, meaning that there is no uncertainty information, USVC degenerates to the SVC solution.

3.2 Extension to Non-linear Case

Generally in real life, data to be classified will require a non-linear separation. USVC needs to be extended to non-linear case. Let $\phi : \mathbb{R}^n \mapsto \mathbb{R}^m$ denote a mapping of the data of input space \mathbb{R}^n to a high dimensional space \mathbb{R}^m . Since the mapped ellipsoid of an uncertain input in the input space may lead to an irregular shape in the feature space, the Taylor Series expansion is introduced and can be expanded based on the inputs \mathbf{x}_i and \mathbf{x}_j . Set $\phi(\mathbf{x}_i) = \mathbf{z}_i = [z_{i1}, \dots, z_{im}]^T \in \mathbb{R}^m$ and $\mathbf{x}_i = [x_{i1}, \dots, x_{in}]^T \in \mathbb{R}^n$, we have $\phi(\mathbf{x}_j) = \phi(\mathbf{x}_i) + \mathbf{J}(\mathbf{x}_j - \mathbf{x}_i) + O\left(\frac{1}{2} \frac{\partial^2 \mathbf{z}}{\partial \mathbf{x}^2} (\mathbf{x}_j - \mathbf{x}_i) + \dots\right)$ where \mathbf{J} is Jacobian matrix which is made up of the first order partial derivatives and can be used to approximately map a tiny line segment $\|\mathbf{x}_i - \mathbf{x}_j\|$ in the input space to $\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|$ in the feature space. The Taylor series of $\phi(\mathbf{x}_j)$ can be simplified by ignoring the higher order partial derivatives, we have $\Delta\phi(\mathbf{x}_j) \simeq \mathbf{J}\Delta\mathbf{x}_j$. Furthermore, the expression can be extended to accommodate the geometric polygonal mapping of the input space, $\left[\Delta\phi(\mathbf{x}_1)^T \quad \dots \quad \Delta\phi(\mathbf{x}_l)^T \right]^T = \left[\Delta\mathbf{x}_1^T \quad \dots \quad \Delta\mathbf{x}_l^T \right]^T \mathbf{J}^T$, where $\Delta\phi(\mathbf{x}_i) \in \mathbb{R}^m$ and $\Delta\mathbf{x}_i \in \mathbb{R}^n$. The covariance matrix \mathbf{M}_i represents the i th uncertainty distribution of the input space. The tiny line segment $\Delta\mathbf{x}_i$ is related to $O(\mathbf{M}_i^{1/2})$, which can be represented as $\mathbf{M}_i = (\mathbf{M}_i^{1/2})^T \mathbf{M}_i^{1/2} \sim \Delta\mathbf{x}_i \Delta\mathbf{x}_i^T$. Therefore, the related geometric mapping of \mathbf{M}_i in the feature space can be formed by the Jacobian matrix

$$\phi(\mathbf{M}_i^{1/2}) = \mathbf{M}_i^{1/2} \mathbf{J}^T \quad (12)$$

According to the definition $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$, $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ can be seen as independent functions during the derivatives over the kernel function,

so the first and second derivatives of the kernel function can be retrieved by the inner product of the mapping function ϕ and its derivative. Therefore, the optimization problem of USVC is given by:

$$\begin{aligned} \max_{\alpha, \beta} \quad & \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{2} \left(\right. \\ & \sum_{i=1}^l \sum_{j=1}^l \alpha_i y_i \left[\frac{\partial K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j} \right]^T (\mathbf{M}_j^{1/2})^T \beta_j + \sum_{i=1}^l \sum_{j=1}^l \alpha_j y_j \beta_i^T \mathbf{M}_i^{1/2} \frac{\partial K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} \\ & \left. + \sum_{i=1}^l \sum_{j=1}^l \beta_i^T \mathbf{M}_i^{1/2} \frac{\partial^2 K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i \partial \mathbf{x}_j} (\mathbf{M}_j^{1/2})^T \beta_j \right) \\ \text{s.t.} \quad & \sum_{i=1}^l \alpha_i y_i = 0 \quad \|\beta_i\| \leq r \alpha_i \quad r \geq 0 \quad 0 \leq \alpha_i \leq C \quad i = 1, \dots, l \end{aligned} \quad (13)$$

where $\frac{\partial K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = \frac{\partial \phi(\mathbf{x}_i)}{\partial \mathbf{x}_i} \cdot \phi(\mathbf{x}_j)$, $\left[\frac{\partial K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j} \right]^T = \phi(\mathbf{x}_i) \cdot \frac{\partial \phi(\mathbf{x}_j)}{\partial \mathbf{x}_j}$ and $\frac{\partial^2 K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i \partial \mathbf{x}_j} = \frac{\partial \phi(\mathbf{x}_i)}{\partial \mathbf{x}_i} \cdot \frac{\partial \phi(\mathbf{x}_j)}{\partial \mathbf{x}_j}$.

4 Minimax Probability Support Vector Classification

4.1 Total Support Vector Classification

In 2004, [2] proposed a formulation of support vector classification called total support vector classification (TSVC), which can accommodate uncertainties in the inputs. Without loss of generality, $\delta \|\mathbf{w}\|$ in [2] can be transformed to $\|\mathbf{M}_i^{1/2} \mathbf{w}\|$ in this paper under the definition of the uncertain inputs in Definition 1. As the result, the constraints of the problem of TSVC becomes to

$$-\|\mathbf{M}_i^{1/2} \mathbf{w}\| \leq y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i. \quad (14)$$

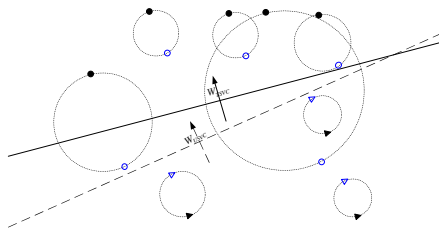


Fig. 2. Geometric Interpretation of TSVC and USVC.

Equation (14) actually can be derived from (6) with $r = -1$. Figure 2 shows the original figure from [2] with the USVC solution at the setting of $r = 1$ superimposed to illustrate the different geometric interpretation between TSVC and USVC. In Fig. 2, the ellipses with circle points on and the ellipses with triangle points on represent the examples from two different classes where $y_i = \pm 1$. In the classification,

TSVC uses the farthest points (solid points) in the distributions of the uncertain inputs as a reference to obtain the optimal hyperplane (\mathbf{w}_{TSVC} , solid

line), while USVC uses the nearest points (hollow points) in the distributions of the uncertain inputs to the optimal hyperplane (\mathbf{w}_{USVC} , dashed line) to compute the classifier. According to the characteristics of support vectors and convex optimization, it can be proved that TSVC is neither a convex optimization nor a SOCP problem.

4.2 Adaptive Constraints in Uncertainty Support Vector Classification

Although TSVC is not a convex optimization, input uncertainty classification can benefit from the characteristics of its low probability confidence. An iterative algorithm is proposed here to combine TSVC and USVC to achieve a better overall performance. This new method is termed adaptive uncertainty support vector classification (AUSVC), in which, the misclassified inputs decrease their probability confidence to accommodate the misclassification in each step, while the probability confidence of correctly classified inputs remain unvaried. Therefore, individual probability confidence r_i is chosen here for each uncertainty instead of selecting a general probability confidence r for all uncertainties in USVC. In order to remain convex in AUSVC, the optimization problem selects the probability confidence $r_i \geq 0$ and $r \in \mathbb{R}$, so that the probability confidence of some misclassified inputs finally achieve 0 when AUSVC converges. Geometrically, AUSVC is a method of searching the optimal points from the nearest points to the central points of different input uncertainties respectively.

The optimization problem of AUSVC can be rewritten from (13) by simply replacing $\|\beta_i\| \leq r\alpha_i$ with $\|\beta_i\| \leq r_i\alpha_i$. Its iterative algorithm is shown below:

Algorithm 2. AUSVC

Initialize $r_i = 1$, $i = 1, \dots, l$, repeat the following three steps until $r_{inew} = r_i$, $i = 1, \dots, l$:

1. Fix r_i , $i = 1, \dots, l$ to the current value, solve (13) for the parameters α_j , β_j , and b ;
2. Substitute the obtained parameters α_j , β_j , b and the training inputs (\mathbf{x}_i, y_i) into

$$g(\mathbf{x}_i, y_i) = \text{sgn} \left[y_i \left(\sum_{j=1}^l \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) + \sum_{j=1}^l \beta_j^T \mathbf{M}_j^{1/2} \frac{\partial K(\mathbf{x}_j, \mathbf{x}_i)}{\partial \mathbf{x}_j} + b \right) \right] \quad (15)$$

respectively to determine whether the inputs are misclassified, $g(\mathbf{x}_i, y_i) < 0$ or correctly classified, $g(\mathbf{x}_i, y_i) \geq 0$. If correctly classified, their probability confidence r_i remain unchanged, otherwise, a predefined positive scalar (normally 0.1) is deducted from its probability confidence r_i , the changed probability confidence is saved in r_{inew} ;

3. If $r_{inew} = r_i$, the optimal results of α_i , β_i , and b are achieved, otherwise, $r_i = r_{inew}$ and return to step 1;

4.3 Minimax Probability Support Vector Classification

MPM not only derives a discriminative method to classify inputs without prior knowledge of the distributions of inputs, but also as the result introduces a measure to compare the different algorithms. This measure is named minimax probability error (MPE), which adds up together the possible maximal misclassified probability of every input uncertainty, which is $\sup_{\mathbf{z}_i \sim (\mathbf{x}_i, \mathbf{M}_i)} \Pr\{y_i(\mathbf{w}^T \mathbf{z}_i + b) \leq 0\} = \frac{1}{1+d_i^2}$, and $d_i^2 = \inf_{y_i(\mathbf{w}^T \mathbf{z}_i + b) \leq 0} (\mathbf{z}_i - \mathbf{x}_i)^T \mathbf{M}_i^{-1} (\mathbf{z}_i - \mathbf{x}_i)$. To incorporate $r_i < 0$ into the optimization problem to provide lower probability confidence, we introduce MPE to generate a new optimization problem as follows:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \sum_{i=1}^l \frac{1}{1+d_i^2} \\ \text{s.t. } \quad & d_i^2 = \begin{cases} \frac{(\mathbf{w}^T \mathbf{x}_i + b)^2}{\mathbf{w}^T \mathbf{M}_i \mathbf{w}} & y_i(\mathbf{w}^T \mathbf{x}_i + b) > 0 \\ 0 & y_i(\mathbf{w}^T \mathbf{x}_i + b) \leq 0 \end{cases} \end{aligned} \quad (16)$$

Geometrically, when the uncertainty is misclassified by a hyperplane, then $d_i = 0$ and its possible maximal misclassified probability on this hyperplane is 1. Otherwise, d_i is equal to the distance between the center \mathbf{x}_i and the hyperplane, and its possible maximal misclassified probability on this hyperplane is $\frac{1}{1+d_i^2}$.

Since this approach is motivated by MPM and SVC, we call this proposed algorithm minimax probability support vector classification (MPSVC). However, the contribution of the maximal misclassified probability of each input is different in the optimization problem. Inspired by [6], in which the prior probability of each class is introduced in the optimization when the worst-case accuracies for two classes are not the same, additional parameters θ_i are introduced with different values set adaptively for the inputs misclassified and correctly classified to formulate a cost sensitive optimization problem. Introducing the previous results from USVC, $\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i + \sum_{i=1}^l (\mathbf{M}_i^{1/2})^T \boldsymbol{\beta}_i$, and the previous results from (6) into (16), kernel functions can be exploited to extend MPSVC to non-linear input uncertainty classification. Equation (16) can be rewritten as:

$$\begin{aligned} \max_{\alpha_i, \boldsymbol{\beta}_i, b, r_i} \quad & \sum_{i=1}^l \theta_i r_i \\ \text{s.t. } \quad & y_i \left(\sum_{j=1}^l \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) + \sum_{j=1}^l \boldsymbol{\beta}_j^T \mathbf{M}_j^{1/2} \frac{\partial K(\mathbf{x}_j, \mathbf{x}_i)}{\partial \mathbf{x}_j} + b \right) \geq r_i \\ & \left\| \sum_{j=1}^l \alpha_j y_j \mathbf{M}_i^{1/2} \frac{\partial K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} + \sum_{j=1}^l \mathbf{M}_i^{1/2} \frac{\partial^2 K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i \partial \mathbf{x}_j} (\mathbf{M}_j^{1/2})^T \boldsymbol{\beta}_j \right\| \leq C \\ & r_i \geq D_i \quad i = 1, \dots, l \end{aligned} \quad (17)$$

where $\theta_i \in \mathbb{R}$ are penalty coefficients, C is a constant, $r_i \in \mathbb{R}$ is the probability confidence of the i th uncertainty, the lower bound of r_i is provided by $D_i \in \mathbb{R}$,

which can be set positive or negative depending on the uncertain inputs and the optimal hyperplane. For those which are misclassified by the hyperplane, D_i will be set negative to decrease the probability confidence of those inputs, otherwise, D_i will be positive. In general, D_i extends the geometric search area of the optimal solution from $r_i \geq 0$ in AUSVC to both $r_i \geq 0$ and $r_i < 0$ in MPSVC while this optimization problem remains a convex and SOCP problem. MPSVC provides a way of searching input uncertainty classification solutions in a specific scope by implementing the optimal solution of AUSVC as additional constraints. Its iterative algorithm is shown below:

Algorithm 3. MPSVC

When MPE converges, MPSVC achieves its optimum.

1. Run Algorithm 2 of AUSVC for the parameters α_j , β_j , and b ;
2. Substitute the obtained parameters α_j , β_j , b and the training inputs (\mathbf{x}_i, y_i) into (15) to determine whether the inputs (\mathbf{x}_i, y_i) are misclassified. If correctly classified, their D_i are set to equal to the average value of d_i of these correctly classified inputs, otherwise, $D_i = -1$, use (16) to calculate MPE_{old} ;
3. Fix $\theta_i = 1$, $i = 1, \dots, l$, solve (17) for the parameters α_j , β_j , and b ;
4. Substitute the parameters α_j , β_j , b and the training inputs (\mathbf{x}_i, y_i) into

$$h(\mathbf{x}_i, y_i) = \text{sgn} \left[y_i \left(\sum_{j=1}^l \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) + \sum_{j=1}^l \beta_j^T \mathbf{M}_j^{1/2} \frac{\partial K(\mathbf{x}_j, \mathbf{x}_i)}{\partial \mathbf{x}_j} + b \right) - 1 \right] \quad (18)$$

to determine whether the inputs (\mathbf{x}_i, y_i) are misclassified ($h(\mathbf{x}_i, y_i) < 0$). If correctly classified, $\theta_i = \theta_i / \lambda$, otherwise, $\theta_i = \theta_i \times \lambda$, λ is a scalar (normally 10);

5. Use (16) to calculate MPE_{new} . If $\|\text{MPE}_{\text{new}} - \text{MPE}_{\text{old}}\| < \epsilon$, MPE converges, otherwise, $\text{MPE}_{\text{old}} = \text{MPE}_{\text{new}}$, return to step 4;

5 Experimental Comparisons

Besides the traditional measure, the number of misclassified centers of the uncertainties (NMC) and the new introduced measure MPE, a new parameter which measures the number of misclassified nearest edges of the uncertainties to the optimal hyperplane is introduced as an additional performance measure in the experiments. This new measure is named as the number of misclassified edges of the uncertainties (NME). NME provides a new viewpoint which includes the uncertainties in the performance comparison of different algorithms.

We reproduced the experiments from [2] to test the performance of these algorithms by following the exact prescription described in [2]. In the experiments with toy data sets in two dimensions, $l = 100$ training examples \mathbf{x}_i were generated from the uniform distribution on $[-5, 5]^2$ by the random number generator. Binary classification problems were created with original separating function $x_1^2 + x_2^2 = Ra^2$, where $Ra \in [3, 4]$ and $\mathbf{x} = [x_1, x_2]^T$. All the algorithms were

trained with the quadratic kernel $(\mathbf{x}_i^T \mathbf{x}_j)^2$ in the experiments. The input vectors \mathbf{x}_i were contaminated by Gaussian noise with mean $[0, 0]$ and covariance matrix $\Sigma = \delta_i \mathbf{I}$ where δ_i was randomly chosen from $[0.1, 0.8]$. \mathbf{I} is a 2×2 identity matrix. We randomly chose $0.1l$ from the first $0.2l$ examples after examples were ordered in an ascending order of their distances to the original separating hyperplane. For these $0.1l$ examples, noise was generated using a larger δ_i randomly drawn from $[0.5, 2]$. In total 16 input datasets are generated with the results of the classification of the 6th and 14th dataset shown in Fig. 3 and the 8th and 12th dataset shown in Fig. 4. The experimental code is based on the MATLAB SVM toolbox [5] and MATLAB optimization toolbox SeDuMi [9].

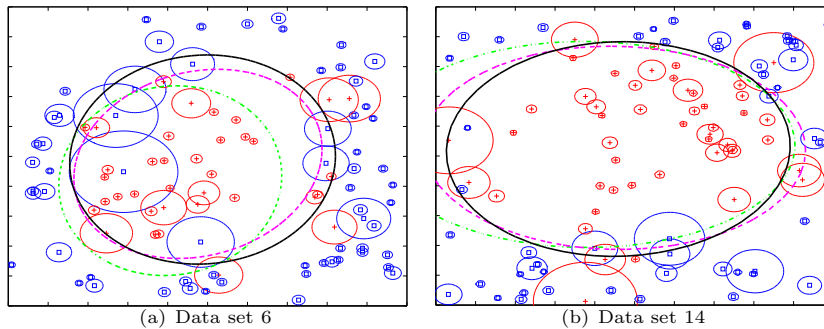


Fig. 3. Experimental comparisons. The dashed line represents USVC ($r = 1$), the dash-dot line represents AUSVC and MPSVC is represented by the solid line.

Figure 3 shows the experimental comparison of 6th and 14th dataset between USVC, AUSVC and MPSVC. Both AUSVC and MPSVC perform better than USVC under all the measures. However, AUSVC can be easily influenced at the area with low input density (see Fig. 3(a)) or the area where one class dominates the other class (see Fig. 3(b)). With the advantages of its characteristics, MPSVC can recover from the adversarial distribution introduced by uncertain inputs.

Because the probability confidence is relatively low from the strategy of the iterative algorithms in the optimization problems, AUSVC and MPSVC generally outperform in the experiments with respect to the other methods, especially when the large uncertainties from one class cross the original boundary to dominate the areas of low input density of the other class. USVC performs worse than the other methods by choosing the nearest edges of these dominant uncertainties geometrically which causes the optimal hyperplane of USVC to be stretched to these dominant uncertainties (see Fig. 4(a)). On the contrary, choosing the farthest edges of the uncertainties geometrically also causes some mistakes in the classification. In Fig. 4(d), USVC performs better than TSVC in areas of low input density of both classes. With the iterative algorithm and individually decreasing probability confidence r_i , AUSVC can generally achieve an improved performance by reducing the influence from some dominant uncertainties in the

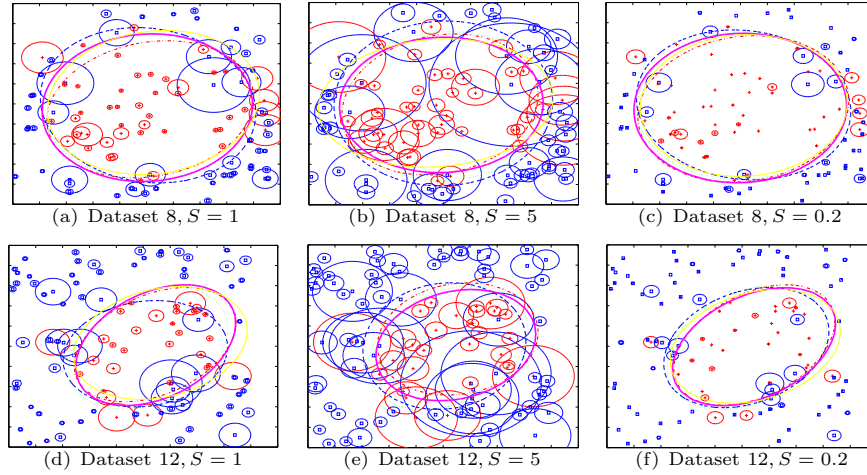


Fig. 4. Experimental Comparisons. Different algorithms are compared in the experiments. The covariance matrices \mathbf{M}_i are varied as $\mathbf{M}'_i = S\mathbf{M}_i$, and the inputs \mathbf{x}_i are kept fixed. The dotted line represents SVC, the dashed line represents TSVC, the thin solid line represents USVC ($r = 1$), the dash-dot line represents AUSVC and MPSVC is represented by thick solid line.

classification (see Fig. 4(a) 4(d)). With lower probability confidence and larger penalty coefficients for misclassified uncertainties, MPSVC can even achieve a better performance than AUSVC by recovering from the adversarial distribution introduced by uncertain inputs (see Fig. 4(d)). The experimental comparison of NMC, NME and MPE of these approaches on the 16 datasets are shown in Fig. 5,

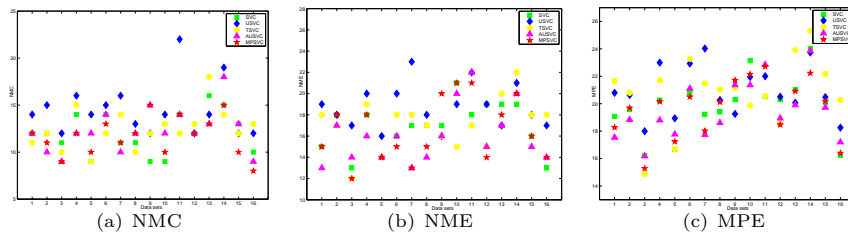


Fig. 5. Experimental comparisons of the different measures of SVC, USVC ($r = 1$), TSVC, AUSVC and MPSVC.

The influence from the uncertainties with different sizes is compared in the experiments as well. Let $S \in \mathbb{R}$ denote the size factor, the varied uncertainties \mathbf{M}'_i come from $\mathbf{M}'_i = S\mathbf{M}_i$. When the uncertainties are amplified by S (see

Fig. 4(b) 4(e), $S = 5$), the performance of USVC deteriorates in Fig. 4(b), and even more in Fig. 4(e), in which USVC can not accommodate large uncertainties during the optimization. AUSVC and MPSVC can produce superior results with its varied probability confidence r_i and penalty coefficients θ_i in iterative algorithms. In the case when the uncertainties decrease (see Fig. 4(c) 4(f), $S = 0.2$), AUSVC achieves the best performance around $r = 1$, which is very close to the performance of USVC. As S decreases, AUSVC and USVC converge to SVC, and in the limit ($S = 0$), the information of the uncertainties is unavailable, AUSVC and USVC degenerate to SVC. In this case, $\beta_i = \mathbf{0}$ and $\|\beta_i\| \leq \alpha_i$ can be rewritten as $\alpha_i \geq 0$ in (13).

6 Conclusions

A new approach, USVC, has been proposed here for classifying data with uncertain information which has been implemented in this paper as additional constraints in the optimization. Along with USVC, two novel iterative algorithms AUSVC and MPSVC have been designed by using adaptive constraints come from the noise-specific covariance information. These methods have been extended to non-linear models by a novel formulation to accommodate kernel functions. Experimental comparisons show that these iterative approaches based around adaptive constraints have greatly improved the performance of input uncertainty classification.

References

1. Bertsimas, D., Popescu, I., Sethuraman, J.: Moment problems and semidefinite optimization. *Handbook of Semidefinite Optimization*. (2000) 469–509
2. Bi, J., Zhang, T.: Support vector classification with input data uncertainty. *Advances in Neural Information Processing Systems*. **16** (2004)
3. Girard, A., Rasmussen, C.E., Quiñero-Candela, J., Murray-Smith, R.: Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting. *Advances in Neural Information Processing Systems*. **15** (2003)
4. Grötschel, M., Lovász, L., Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*. 2nd corr. ed.. Springer-Verlag. ISBN: 0-38-756740-2 (1993) 66–73
5. Gunn, S.R.: *Support vector machines for classification and regression*. Technical Report. University of Southampton. (1998)
6. Huang, K., Yang, H., King, I., Lyu, M.R., Chan, L.: The minimum error minimax probability machine. *Journal of Machine Learning Research*. **5** (2004) 1253–1286
7. Lanckriet, G.R.G., El Ghaoui, L., Bhattacharyya, C., Jordan, M.I.: A robust minimax approach to classification. *Journal of Machine Learning Research*. **3** (2002) 555–582
8. Shivaswamy, P.K., Bhattacharyya, C., Smola, A.J.: Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*. **7** (2006) 1283–1314
9. Sturm, J.F.: Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*. **11-12** (1999) 625–653
10. Yang, J., Gunn, S.R.: *Input uncertainty in support vector machines*. Machine Learning Workshop, Sheffield, UK, (2004)