A new robust model of one-class classification by interval-valued training data using the triangular kernel

Lev V. Utkin*
Department of Control, Automation and System Analysis,
Saint Petersburg State Forest Technical University, Russia
lev.utkin@gmail.com

Anatoly I. Chekh
Department of Computer Science,
Saint Petersburg State Electrotechnical University, Russia
anatoly.chekh@gmail.com

Abstract

A robust one-class classification model as an extension of Campbell and Bennett's (C-B) novelty detection model on the case of interval-valued training data is proposed in the paper. It is shown that the dual optimization problem to a linear program in the C-B model has a nice property allowing to represent it as a set of simple linear programs. It is proposed also to replace the Gaussian kernel in the obtained linear support vector machines by the well-known triangular kernel which can be regarded as an approximation of the Gaussian kernel. This replacement allows us to get a finite set of simple linear optimization problems for dealing with interval-valued data. Numerical experiments with synthetic and real data illustrate performance of the proposed model.

Keywords: one-class classification, novelty detection, support vector machine, kernel, interval-valued data, minimax strategy, linear programming, extreme points

1 Introduction

One of the problems of the statistical machine learning is to classify some objects into classes in accordance with their properties or features. At the same time, we need often to detect abnormal examples or to solve an one-class classification (OCC) or novelty detection problem. A lot of papers are devoted to this

*Corresponding author

important problem [6, 7, 14, 28, 37, 39, 50]. Various reviews of the OCC can be found in the machine learning literature, for example, reviews provided by Markou and Singh [29], by Bartkowiak [2], by Khan and Madden [25], by Hodge and Austin [23]. The OCC aims to detect anomalous or abnormal observations and separate them from the so-called normal examples [10, 11, 43].

A common way for solving the OCC problem is to model the support of the unknown data distribution directly from data, that is, to estimate a binary-valued function f that is positive in a region where the density is high, and negative elsewhere. Sample points outside this region can be regarded as anomalous observations.

Some models of the OCC are based on using the framework of the support vector machine (SVM). These models are called OCC SVMs. We mark out three main approaches for constructing the OCC SVMs. The first approach is proposed by Tax and Duin [44, 45]. This is one of the well-known OCC models, which can be regarded as an unsupervised learning problem. According to this approach, the training of the one-class SVM consists in determining the smallest hyper-sphere containing training data. An alternative way to geometrically enclose a fraction of the training data is via a hyperplane and its relationship to the origin proposed by Schölkopf at el. [37, 39]. Under this approach, a hyperplane is used to separate the training data from the origin with the maximal margin, i.e., the objective is to separate off the region containing the data points from the surface region containing no data. It should be noted that both the approaches provide the same results when a symmetric kernel is used. The third approach which will be considered in detail in the paper is the linear programming approach to the OCC proposed by Campbell and Bennett [7]. The model proposed by Campbell and Bennett uses linear programming techniques.

It should be noted that there are other interesting novelty detection or OCC models (see for instance, [5, 23, 26, 27]). Every model can be applied in various applications.

All these OCC models are based on using a training set consisting of precise or point-valued data. However, training examples in many real applications can be obtained only in the interval form. Interval-valued data may result from imperfection of measurement tools or imprecision of expert information. There may also be some missing data when some features of an example are not observed [34].

Many methods in machine learning have been presented for dealing with interval-valued data [24, 32, 41] due to the importance of this condition. In some methods, interval-valued observations are replaced by precise values based on some additional assumptions, for example, by taking middle points of intervals [31]. This approach can be successfully used when intervals are not large and the area produced by the interval intersections is rather small (see, for example, the left picture in Fig. 1). However, if the intervals are very large (see, for example, the right picture in Fig. 1), then the replacement of intervals by point-valued data may lead to large classification errors.

Another part of methods use the standard interval analysis for constructing the classification and regression models [1, 22]. A series of interesting models

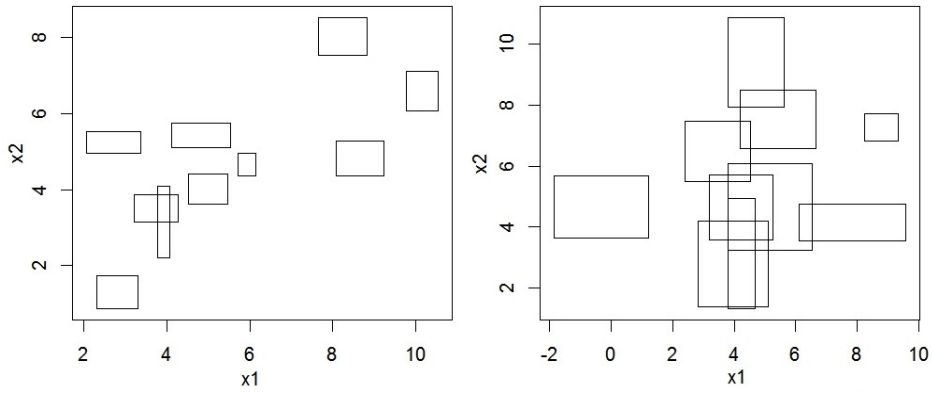

Figure 1: Examples of interval-valued data with small and large intervals

for dealing with interval-valued and fuzzy observations in classification and regression can be found in works [8, 9, 19]. However, these models as well as the standard interval analysis are restricted by considering only the linear case, i.e., when separating or regression functions are linear.

Do and Poulet [16] proposed an interesting and very simple method based on the change of the Euclidean distance between two data points in the Gaussian kernel function by the Hausdorff distance between two hyper-rectangles produced by intervals from sample data. The method can be used in classification and regression analyses, in OCC problems. The main condition of its use is the assumption of the Gaussian kernel (or the kernels based on the distance between points) in the corresponding SVM. In spite of its simplicity, the method has an important obstacle for its application. It is not known how to interpret the classification results. Moreover, by dealing with interval-valued data, we usually implicitly or explicitly select a point in every interval in accordance with some decision strategy, which can be regarded as a "typical" point of the interval under the accepted decision strategy. The method using the Hausdorff distance allows having many different data points in intervals simultaneously, namely, pairwise distances between three intervals may correspond to different points in every interval. Another disadvantage of the approach based on using the Hausdorff distance is a lack of some justified strategy of decision making by dealing with imprecise data. In other words, it is not obvious in using the Hausdorff distance how to interpret the points which determine the distance between intervals from the classification point of view. The Hausdorff distance also was used in clustering with imprecise data, for example, Chavent et al. [12, 13] proposed a partitional dynamic clustering method for interval data based on adaptive Hausdorff distances. A city-block distance function as the distance of a special form for solving clustering problems under interval-valued data was studied by de Souza and de Carvalho [15]. Pedrycz et al. [33] exploited a concept of the Hausdorff distance that determines a distance between some in-

formation granule and a numeric pattern (a point in the highly dimensional feature space) for constructing classifiers by interval and fuzzy data. It should be noted that other distance measures have been successfully applied to machine learning problems. For example, Schollmeyer and Augustin [40] proposed another distance measure for solving regression problems under interval data. The authors [40] argued that their measure might be better in some problems because the Hausdorff distance does not match points of two sets but compares all points of the two sets to each other.

Another interesting approach to constructing a classifier under interval-valued data was proposed by Bhadra et al. [4]. The authors presented a novel methodology using Bernstein bounding schemes for constructing classifiers which are robust to interval-valued uncertainty in examples. According to the methodology, the uncertain examples are classified correctly with high probability. A binary linear classification model under interval data different from the models using the point-valued representation of intervals was proposed by El Ghaoui et al. [21]. The authors develop a robust classifier by minimizing the worst-case value of a given loss function over all possible choices of the data in the multi-dimensional intervals.

Following the idea of the robust model provided by El Ghaoui et al. [21], we propose a robust model which is based on three main ideas implemented in order to construct a new OCC model dealing with interval-valued training data.

1. Interval-valued observations produce a set of expected classification risk measures such that the lower and upper risk measures can be determined by minimizing and by maximizing the risk measure over values of intervals.

2. There are many variants of OCC SVMs. It is proposed to use linear programming OCC SVM by Campbell and Bennett [7] for which constraints in its dual form do not depend on vectors of observations. This allows us to represent the dual optimization problem as a set of simple optimization problems.

3. It is proposed to replace the Gaussian kernel by the well-known triangular kernel which can be regarded as an approximation of the Gaussian kernel. This replacement allows us to get a set of linear optimization problems with variables $\mathbf{x}_i$ restricted by intervals $\mathbf{A}_i$, $i = 1, ..., n$.

The triangular kernel has been used by Utkin et al. [47] as an approximation of the Gaussian kernel in the framework of SVM based on Schölkopf approach to model the OCC. However, the main difficulty in using the proposed method is extremely hard computations in order to enumerate all vertexes of hyper-rectangles produced by interval-valued data especially when training examples have a lot of interval-valued features. In the present paper, we also apply the triangular kernels to approximate the Gaussian kernel. But the advantage of the proposed model is that its computational complexity does not substantially depends on the number of features. At the same time, we cannot assert that the

model is rather simple from the computation point of view when the number of observations is large.

It is important to point out that the proposed model eventually deals with some points in interval-valued data which can be called optimal to some extent. However, in contrast to the models where intervals are replaced by points, the proposed model searches for optimal precise points by applying the robust or maximin strategy. In fact, we select a single probability distribution or a point in the interval of expected risk values in accordance with a certain decision strategy instead of points in intervals of training data.

The paper is organized as follows. Section 2 presents a short introduction into the Campbell-Bennett novelty detection model proposed in [7]. An approach for extension of the Campbell-Bennett model on the case of interval-valued data is provided in Section 3. Numerical experiments with synthetic and real data illustrating accuracy of the proposed algorithm are given in Section 4. In Section 5, concluding remarks are made.

2 Campbell and Bennett's novelty detection linear model

Suppose we have unlabeled training data $\mathbf{x}_1, ..., \mathbf{x}_n \subset \mathcal{X}$, where n is the number of observations, $\mathcal{X}$ is some set, for instance, it is a compact subset of $\mathbb{R}^l$. Let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ be drawn i.i.d. from a distribution on $\mathcal{X}$. The sample space $\mathcal{D}$ is finite and discrete. According to papers [37, 39], a well-known novelty detection or OCC model aims to construct a function f which takes the value $+1$ in a "small" region capturing most of the data points and -1 elsewhere. It can be done by mapping the data into the feature space corresponding to a kernel and by separating them from the origin with the maximum margin.

Let ϕ be a feature map $\mathcal{X} \to G$ such that the data points are mapped into an alternative higher-dimensional feature space G. In other words, this is a map into an inner product space G such that the inner product in the image of ϕ can be computed by evaluating some simple kernel $K(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x}), \phi(\mathbf{y}))$, such as the Gaussian kernel

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\left\|\mathbf{x} - \mathbf{y}\right\|^2 / \gamma^2\right).$$

Here γ is the kernel parameter determining the geometrical structure of the mapped samples in the kernel space. It is pointed out by [49] that the problem of selecting a proper parameter γ is very important in classification. When a very small γ is used ($\gamma \to 0$), $K(\mathbf{x}, \mathbf{y}) \to 0$ for all $\mathbf{x} \neq \mathbf{y}$ and all mapped samples tend to be orthogonal to each other, despite their class labels. In this case, both between-class and within-class variations are very large. On the other hand, when a very large γ is chosen ($\gamma^2 \to \infty$), $K(\mathbf{x}, \mathbf{y}) \to 1$ and all mapped samples converge to a single point. This obviously is not desired in a classification task. Therefore, a too large or too small γ will not result in more separable samples in G.

We consider the linear programming approach to novelty detection proposed by Campbell and Bennett [7]. The authors start from the hard margin case, when any training point $\mathbf{x}_j$ lying outside some predefined surface restricted the training points is viewed as abnormal. This surface is defined as the level set, $f(\mathbf{z}) = 0$, of some nonlinear function. In feature space, $f(\mathbf{z}) = \sum_i \varphi_i K(\mathbf{z}, \mathbf{x}_i) + b$, this corresponds to a hyperplane which is pulled onto the mapped data points with the restriction that the margin always remains positive or zero [7]. Here $\varphi = (\varphi_n, ..., \varphi_n)$ are parameters of the function f in the feature space or Lagrange multipliers.

A criteria for constructing the optimal function $f(\mathbf{z})$ proposed by Campbell and Bennett is to minimize the mean value of the output of the function, i.e., $\sum_i f(\mathbf{x}_i)$. This is achieved by minimizing:

$$W(\varphi, b) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \varphi_j K(\mathbf{x}_i, \mathbf{x}_j) + b \right),$$

subject to

$$\sum_{j=1}^{n} \varphi_j K(\mathbf{x}_i, \mathbf{x}_j) + b \geq 0, \ i = 1, ..., n,$$

$$\sum_{i=1}^{n} \varphi_i = 1, \ \varphi_i \geq 0. \tag{1}$$

The bias b is just treated as an additional parameter in the minimization process. The added constraints on φ restrict the class of models to be considered. As indicated by Campbell and Bennett [7], these constraints amount to a choice of scale for the weight vector normal to the hyperplane in feature space and hence do not impose a restriction on the model. Also, these constraints ensure that the problem is well-posed and that an optimal solution with $\varphi \neq 0$ exists. Other constraints on the class of functions are possible, e.g. $\|\varphi\|_1 = 1$ with no restriction on the sign of φ_i.

It is important to point out here that Campbell and Bennett propose to use the mean value of the output of the function. It follows from the form of $W(\varphi, b)$ that the empirical probability distribution $(1/n, ..., 1/n)$ is assumed to get the mean value $W(\varphi, b)$. The multiplier $1/n$ is omitted because it does not change the optimization variables φ and b.

To handle noise and outliers a soft margin is introduced in analogy to the usual approach used with support vector machines [14, 38, 42, 48]. In this case, the following function has to be minimized:

$$W(\varphi, b) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \varphi_j K(\mathbf{x}_i, \mathbf{x}_j) + b \right) + \frac{1}{vn} \sum_{i=1}^{n} \xi_i, \tag{2}$$

subject to (1) and

$$\sum_{j=1}^{n} \varphi_j K(\mathbf{x}_i, \mathbf{x}_j) + b \geq -\xi_i, \ \xi_i \geq 0, \ i = 1, ..., n. \tag{3}$$

The parameter $v \in [0;1]$ controls the extent of margin errors (smaller v means fewer outliers are ignored: $v \to 0$ corresponds to the hard margin limit). It is a parameter which is analogous to ν for the ν-SVM standard method [14]. Slack variables $\xi = (\xi_1, ..., \xi_n)$ are used to allow points to violate margin constraints.

We will shortly call Campbell and Bennett's novelty detection model below as the C-B model.

It should be noted that $W(\varphi, b)$ in (2) can be viewed as the expected risk measure. This is important for the next consideration of interval-valued data.

3 C-B model and interval-valued data

Let us consider how C-B model can be modified under condition that examples from the training set are interval-valued. We consider classification problems where the input variables (patterns) $\mathbf{x}$ may be interval-valued. Suppose that we have a training set $(\mathbf{A}_i)$, $i = 1, ..., n$. Here $\mathbf{A}_i \subset \mathbb{R}^m$ is the Cartesian product of m intervals $[\underline{a}_i^{(k)}, \overline{a}_i^{(k)}]$, $k = 1, ..., m$, which again are not restricted so could even include intervals $(-\infty, \infty)$. In other words, every feature of every observation or training example is interval-valued. We aim to construct a function f which takes the value $+1$ in a "small" region capturing most of the interval-valued examples and -1 elsewhere.

Suppose that every set of points $\mathbf{x}_1, ..., \mathbf{x}_n$ belonging to $\mathbf{A}_1, ..., \mathbf{A}_n$, respectively, produces a decision function $f(\mathbf{x})$ by means of C-B model, i.e., by solving the linear programming problem (2)-(3). All possible combinations of points $\mathbf{x}_1, ..., \mathbf{x}_n$ from sets $\mathbf{A}_1, ..., \mathbf{A}_n$ produce a set of decision functions. Our first aim is to define a single function from the set of decision function corresponding to a certain robust decision strategy.

The robust classifier in this case is the one that maximizes the expected risk measure. The robust strategy in this case can be also viewed as a minimax strategy which selects the "worst" combination of points from intervals $\mathbf{A}_1, ..., \mathbf{A}_n$ providing the largest value of the expected risk. The minimax strategy can be interpreted as an insurance against the worst case because it aims at minimizing the expected loss in the least favorable case [35].

Using the pessimistic strategy, we can write the optimization problem as follows:

$$W(\varphi, b) = \sup_{\mathbf{x}_i \in \mathbf{A}_i, i=1,...,n} \min_{\varphi, b, \xi} \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n} \varphi_j K(\mathbf{x}_i, \mathbf{x}_j) + b \right) + \frac{1}{v} \sum_{i=1}^{n} \xi_i \right), \quad (4)$$

subject to (1) and (3).

In order to solve the above optimization problem, we sketch the following steps:

1. We fix the values $\mathbf{x}_1, ..., \mathbf{x}_n$ and write a set of dual forms of problem (4).

2. We solve every dual problem by means of extreme points which do not depend on $\mathbf{x}_1, ..., \mathbf{x}_n$.

3. We reduce the optimization problems with sets of values $\mathbf{x}_1, ..., \mathbf{x}_n$ to the linear ones by introducing new kernels.

4. The set of linear problems with variables $\mathbf{x}_1, ..., \mathbf{x}_n$ are solved by means of standard methods.

Below every step is represented as a subsection.

3.1 A set of dual optimization problems

Below we show that the initial optimization problem (4) can be represented as a finite set of simplified programming problems.

Let us fix values $\mathbf{x}_1, ..., \mathbf{x}_n$ in (1), (3) and (4). Then we get a linear programming problem with variables $\varphi = (\varphi_n, ..., \varphi_n)$, $\xi = (\xi_1, ..., \xi_n)$, b for every fixed values $\mathbf{x}_1, ..., \mathbf{x}_n$. It can be written in the matrix form as follows:

$$\min_{\psi} (\mathbf{c}\psi),$$

subject to $\mathbf{A}\psi \geq \mathbf{q}$.

Here $\psi = (\varphi, \xi, b_0, b_1)$ is the vector of $2n + 2$ non-negative optimization variables. The unconstrained variable b is replaced by two non-negative variables b_0 and b_1. Elements of the vector $\mathbf{c}$ are

$$c_j = \begin{cases} \sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j), & j = 1, ..., n, \\ v^{-1}, & j = n + 1, ..., 2n, \\ n, & j = 2n + 1, \\ -n, & j = 2n + 2. \end{cases}$$

Elements of the first and the second rows of the matrix $\mathbf{A}$ correspond to constraint (1) and are

$$a_{1j} = \begin{cases} 1, & j = 1, ..., n, \\ 0, & j = n + 1, ..., 2n + 2, \end{cases}$$

$$a_{2j} = \begin{cases} -1, & j = 1, ..., n, \\ 0, & j = n + 1, ..., 2n + 2. \end{cases}$$

Other n rows of the matrix $\mathbf{A}$ consist of the matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]_{n \times n}$, the unit matrix $\mathbf{I}_n$ and two elements -1, 1. The vector $\mathbf{q}$ has the first element equal to 1, the second element equal to -1 and n zeros elsewhere.

Let us write the dual form of this linear optimization problem by means of the well-known technique. It can be written in the matrix form as

$$\max_{\phi} (\phi\mathbf{q}),$$

subject to $\phi\mathbf{A} \leq \mathbf{c}^{\mathrm{T}}$.

Here ϕ is the vector of $n+2$ non-negative optimization variables such that $\phi = (c, d, \beta_1, ..., \beta_n)$, where $(\beta_1, ..., \beta_n) = \beta$ is the vector of non-negative optimization variables, c and d are also non-negative optimization variables. After substituting the elements of $\mathbf{A}$, $\mathbf{q}$ and $\mathbf{c}$ into the above dual problem, we can rewrite it as follows:

$$c - d \to \max_{c,d,\beta},$$

subject to

$$c - d + \sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j)\beta_i \le \sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j),\ j = 1, ..., n,$$

$$\beta_i \le \frac{1}{\nu},\ i = 1, ..., n,\ \sum_{i=1}^{n} \beta_i \le n,\ -\sum_{i=1}^{n} \beta_i \le -n.$$

Finally, we can write

$$c - d \to \max_{d,\beta}, \tag{5}$$

subject to

$$c - d - \sum_{i=1}^{n} (1 - \beta_i) K(\mathbf{x}_i, \mathbf{x}_j) \le 0,\ j = 1, ..., n, \tag{6}$$

$$0 \le \beta_i \le \frac{1}{\nu},\ i = 1, ..., n,\ \sum_{i=1}^{n} \beta_i = n. \tag{7}$$

Let us rewrite constraints (6) as

$$c - d \le \sum_{i=1}^{n} (1 - \beta_i) K(\mathbf{x}_i, \mathbf{x}_j),\ j = 1, ..., n.$$

Note that all above inequalities have to be satisfied. This can be achieved when the left side of every inequality is less than the smallest right side. Hence, we can replace n constraints (6) by the following constraint:

$$c - d \le \min_{j=1,...,n} \sum_{i=1}^{n} (1 - \beta_i) K(\mathbf{x}_i, \mathbf{x}_j).$$

Let us replace variables $\beta_1, ..., \beta_n$ by $\alpha_1, ..., \alpha_n$ such that $\alpha_i = \beta_i / n$. Since there are no other restrictions for $c - d$ in (5)-(7) except for (6), then problem (5)-(7) can be rewritten as a set of n optimization problems

$$\left(\max_{\alpha} \sum_{i=1}^{n} (1 - n\alpha_i) K(\mathbf{x}_i, \mathbf{x}_j) \right) \to \min_{j=1,...,n}, \tag{8}$$

subject to

$$0 \le \alpha_i \le \frac{1}{\nu n},\ i = 1, ..., n,\ \sum_{i=1}^{n} \alpha_i = 1. \tag{9}$$

In other words, we have to solve n optimization problems such that the final solution is determined by choosing the smallest objective function (8). The main advantage of the above optimization problem is that the constraints (9) do not depend on $\mathbf{x}_1, ..., \mathbf{x}_n$ and they are linear. Therefore, the problem solution is defined only by extreme points of a set of variables $\alpha_1, ..., \alpha_n$, which also does not depend on $\mathbf{x}_1, ..., \mathbf{x}_n$. This peculiarity is very important and gives us opportunity to replace the optimization problem by a finite set of unconstrained optimization problems.

3.2 Extreme points of the polytope produced by constraints

It should be noted that every problem (8)-(9) by fixed $\mathbf{x}_1, ..., \mathbf{x}_n$ can be solved by using the extreme points of the polytope produced by constraints (9). Suppose that we have T extreme points denoted as $\alpha^{(1)}, ..., \alpha^{(T)}$. Here $\alpha^{(l)} = (\alpha_1^{(l)}, ..., \alpha_n^{(l)})$. Hence, we can rewrite every problem (8)-(9) as follows:

$$\left(\max_{l=1,...,T} \sum_{i=1}^{n} (1 - n\alpha_i^{(l)}) K(\mathbf{x}_i, \mathbf{x}_j) \right) \to \min_{j=1,...,n}. \tag{10}$$

If we would know point-valued training data $\mathbf{x}_1, ..., \mathbf{x}_n$, then we get nT very simple objective functions without constraints. Let us determine the extreme points and their number.

Proposition 1 *Let $\alpha_1, \ldots, \alpha_n$ with $n \in \mathbb{N}$ be a set of data and $\mathcal{M}_v$ be a set of $(\alpha_1, \ldots, \alpha_n)$ produced by conditions*

$$0 \le \alpha_i \le \frac{1}{vn},\ i = 1, ..., n,\ \sum_{i=1}^{n} \alpha_i = 1.$$

1. *If $v \ge (n-1)n^{-1}$, then the set $\mathcal{M}_v$ has $T = n$ extreme points which are of the following form: the k-th element is given by $v^{-1}(n^{-1} + v - 1)$ and the other $n-1$ elements are equal to $v^{-1}n^{-1}$.*

2. *If $n^{-1} < v < (n-1)n^{-1}$, then the set $\mathcal{M}_v$ has $T = s\binom{n}{s}$ extreme points, where $s \in \mathbb{N}$ and it is defined by the inequality*

$$\frac{1}{n-s+1} \le \frac{1}{vn} \le \frac{1}{n-s}.$$

   The extreme points have the same form: $n-s$ elements have value $v^{-1}n^{-1}$, there is one element given by $1-(n-s)v^{-1}n^{-1}$, and the other $s-1$ elements are equal to zero.

3. *If $v \le n^{-1}$, then $\mathcal{M}_v$ coincides with the unit simplex whose vertices have one element equal to 1 and $n-1$ zero elements equal to zero.*

The proof of a similar proposition can be found in [46]. The above proposition provides a simple way for constructing the set of extreme points $\alpha^{(1)}, ..., \alpha^{(T)}$. We have now nT objective functions which have to be minimized over all $j = 1, ..., n$, and to be maximized over $l = 1, ..., T$ for every j. Of course, the above does not mean that problem (8)-(9) is reduced to a set of objective functions because every problem depends on fixed $\mathbf{x}_1, ..., \mathbf{x}_n$. Therefore, the next task is to consider how problem (8)-(9) can be solved by taking into account the set of values $\mathbf{x}_1, ..., \mathbf{x}_n$.

3.3 The triangular kernel

It was assumed in optimization problem (10) that there is a fixed set of points $\mathbf{x}_1, ..., \mathbf{x}_n$ from intervals $\mathbf{A}_1, ..., \mathbf{A}_n$, respectively. Now we relax this condition and try to solve the optimization problem. The main idea to solve the problem is to replace Gaussian kernel $K(\mathbf{x}, \mathbf{y})$ by a new kernel function

$$T(\mathbf{x}, \mathbf{y}) = \max\{0, 1 - \|\mathbf{x} - \mathbf{y}\|^1 / \gamma^2\}.$$

This is the well-known triangular kernel. It can be regarded as an approximation of the Gaussian kernel. The introduced kernel is bounded by 0 and 1. Its largest value 1 takes place when $\mathbf{x} = \mathbf{y}$. A main peculiarity of the function is that it is linear. This peculiarity allows us to represent optimization problem (10) as a set of linear programming problems of a special form.

It should be noted that the triangular kernel is a conditionally positive definite kernel, but the convergence of SVMs remains guaranteed with this type of kernel [37]. Fleuret and Sahbi in [18] show that triangular kernels have a very interesting property: if both training and testing data are scaled by the same factor, the response of the classification function remains the same. This is the scale-invariance property of SVMs based on the Triangular Kernel. The property has been successfully applied to some real applications (see, for example, papers [17, 30, 36]).

3.4 Constructing the linear programming problem with the triangular kernel

The next step is to show that the use of the triangular kernel leads to a linear programming problem which can be simply solved by means of the available standard methods.

Introduce the optimization variables $G_{ij} = T(\mathbf{x}_i, \mathbf{x}_j)$ and $H_{ij}^{(k)} = \left| x_i^{(k)} - x_j^{(k)} \right|$. Then there holds

$$G_{ij} = \max\left\{0, 1 - \sum_{k=1}^{m} H_{ij}^{(k)} / \gamma^2\right\}.$$

Hence, we can write the following objective function for computing optimal

values of G_{ij}, $H_{ij}^{(k)}$, $\mathbf{x}_i$:

$$\left(\max_{\mathbf{x}_i, G_{ij}, H_{ij}^{(k)}} \sum_{i=1}^{n} (1 - n\alpha_i^{(k)}) G_{ij} \right) \rightarrow \max_{k=1,...,T} \min_{j=1,...,n}.$$

Let us consider two main cases of elements of extreme points $\alpha_i^{(k)}$. Denote $N = \{1, ..., n\}$.

Case 1. For every k, we define a set of indices U_k such that $1 - n\alpha_i^{(k)} < 0$ for every $i \in U_k$. In this case, the maximization of the objective function follows the minimization of G_{ij}. Then we have simple constraints for G_{ij}:

$$G_{ij} \geq 0,\ G_{ij} \geq 1 - \sum_{k=1}^{m} H_{ij}^{(k)} / \gamma^2,\ i \in U_k.$$

Simultaneously, the minimization of G_{ij} follows the maximization of every $H_{ij}^{(k)}$. The main problem here is to introduce the absolute value $\left| x_i^{(k)} - x_j^{(k)} \right|$ into the constraints. In order to realize that we use results proposed by Beaumont [3] represented as a lemma. We give its simplified form.

Lemma 2 (Beaumont [3]) *If $[\underline{x}, \overline{x}] \subset \mathbb{R}$, $\underline{x} < \overline{x}$, and, if*

$$u = \frac{|\overline{x}| - |\underline{x}|}{\overline{x} - \underline{x}}, \quad v = \frac{\overline{x}|\underline{x}| - \underline{x}|\overline{x}|}{\overline{x} - \underline{x}},$$

we have

$$\forall x \in [\underline{x}, \overline{x}], \quad |x| \leq ux + v.$$

Let us determine lower and upper bounds for the difference $x_i^{(k)} - x_j^{(k)}$. Its lower bound is $\underline{x}_{ij}^{(k)} = \underline{a}_i^{(k)} - \overline{a}_j^{(k)}$. The upper bound can be obtained in the same way $\overline{x}_{ij}^{(k)} = \overline{a}_i^{(k)} - \underline{a}_j^{(k)}$. Hence, we get the following constraints in accordance to Lemma 1:

$$H_{ij}^{(k)} \leq \underline{h}_{ij}^{(k)} \left(x_i^{(k)} - x_j^{(k)} \right) + \overline{h}_{ij}^{(k)},\ i \in U_k.$$

where

$$\underline{h}_{ij}^{(k)} = \frac{\left| \overline{x}_{ij}^{(k)} \right| - \left| \underline{x}_{ij}^{(k)} \right|}{\overline{x}_{ij}^{(k)} - \underline{x}_{ij}^{(k)}},\ \overline{h}_{ij}^{(k)} = \frac{\overline{x}_{ij}^{(k)} \left| \underline{x}_{ij}^{(k)} \right| - \underline{x}_{ij}^{(k)} \left| \overline{x}_{ij}^{(k)} \right|}{\overline{x}_{ij}^{(k)} - \underline{x}_{ij}^{(k)}}.$$

Case 2. For every k, we define a set of indices $N \backslash U_k$ such that $1 - n\alpha_i^{(k)} \geq 0$ for every $i \in N \backslash U_k$. In this case, the maximization of the objective function follows the maximization of G_{ij}. Denote the non-zero part of the equality for G_{ij} as w, i.e., $G_{ij} = \max(0, w)$. Rewrite the last expression as follows:

$$\max(0, w) = w/2 + \max(-w/2, w/2) = w/2 + |w/2|.$$

Indeed, if w is negative, then $w/2 + |w/2| = 0$. If w is positive, then $w/2 + |w/2| = w$.

We again use results proposed by Beaumont [3] represented in Lemma 1. Let us determine lower and upper bounds for

$$w_{ij} = 1 - \sum_{k=1}^{m} H_{ij}^{(k)}/\gamma^2.$$

First, we find the bounds for the absolute value $H_{ij}^{(k)}$. Its lower bound is

$$\underline{H}_{ij}^{(k)} = \begin{cases} \min\left(|\underline{x}_{ij}^{(k)}|, |\overline{x}_{ij}^{(k)}|\right), & \underline{x}_{ij}^{(k)} \cdot \overline{x}_{ij}^{(k)} \le 0, \\ 0, & \text{otherwise.} \end{cases}$$

Here $\underline{x}_{ij}^{(k)} = \underline{a}_i^{(k)} - \overline{a}_j^{(k)}$ and $\overline{x}_{ij}^{(k)} = \overline{a}_i^{(k)} - \underline{a}_j^{(k)}$ have been defined above. It can be seen that the lower bound for $H_{ij}^{(k)}$ depends on the relative position of the i-th and j-th intervals. In particular, if $\overline{a}_i^{(k)} < \underline{a}_j^{(k)}$ (intervals are not intersecting and the j-th interval follows the i-th interval), then the closest points of intervals are $\overline{a}_i^{(k)}$, $\underline{a}_j^{(k)}$ and the lower bound is $|\overline{x}_{ij}^{(k)}|$. If $\overline{a}_j^{(k)} < \underline{a}_i^{(k)}$, then the lower bound is $|\underline{x}_{ij}^{(k)}|$. However, when the intervals are intersecting (condition $\underline{x}_{ij}^{(k)} \cdot \overline{x}_{ij}^{(k)} > 0$), then there are points belonging to both intervals such that the difference is 0. This difference is the smallest value because $H_{ij}^{(k)} \ge 0$.

The upper bound can be obtained in the same way

$$\overline{H}_{ij}^{(k)} = \max\left(|\underline{x}_{ij}^{(k)}|, |\overline{x}_{ij}^{(k)}|\right).$$

Finally, the lower and upper bounds for w_{ij} are

$$\underline{w}_{ij} = 1 - \sum_{k=1}^{m} \overline{H}_{ij}^{(k)}/\gamma^2, \quad \overline{w}_{ij} = 1 - \sum_{k=1}^{m} \underline{H}_{ij}^{(k)}/\gamma^2,$$

respectively, and the additional constraint is

$$w_{ij} \le \underline{W}_{ij}\left(1 - \sum_{k=1}^{m} H_{ij}^{(k)}/\gamma^2\right) + \overline{W}_{ij}, \ i \in N \backslash U_k,$$

where

$$\underline{W}_{ij} = \frac{|\overline{w}_{ij}| - |\underline{w}_{ij}|}{\overline{w}_{ij} - \underline{w}_{ij}}, \quad \overline{W}_{ij} = \frac{\overline{w}_{ij}\,|\underline{w}_{ij}| - \underline{w}_{ij}\,|\overline{w}_{ij}|}{\overline{w}_{ij} - \underline{w}_{ij}}.$$

Simultaneously, the maximization of G_{ij} follows the minimization of every $H_{ij}^{(k)}$. This can be simply realized by means of constraints:

$$H_{ij}^{(k)} \ge x_i^{(k)} - x_j^{(k)}, \ H_{ij}^{(k)} \ge x_j^{(k)} - x_i^{(k)}, \ i \in N \backslash U_k.$$

In sum, we get the following optimization problem:

$$O(j,k) = \max_{\mathbf{x}_i, G_{ij}, H_{ij}^{(k)}} \left\{ \sum_{i \in U_k} (1 - n\alpha_i^{(k)}) G_{ij} + \frac{1}{2} \sum_{i \in N \setminus U_k} (1 - n\alpha_i^{(k)}) \left(w_{ij} + 1 - \sum_{k=1}^{m} H_{ij}^{(k)} / \gamma^2 \right) \right\}, \tag{11}$$

subject to

$$G_{ij} \geq 0,\ G_{ij} \geq 1 - \sum_{k=1}^{m} H_{ij}^{(k)} / \gamma^2,\ i \in U_k. \tag{12}$$

$$H_{ij}^{(k)} \leq \underline{h}_{ij}^{(k)} \left(x_i^{(k)} - x_j^{(k)} \right) + \overline{h}_{ij}^{(k)},\ i \in U_k, \tag{13}$$

$$w_{ij} \leq \underline{W}_{ij} \left(1 - \sum_{k=1}^{m} H_{ij}^{(k)} / \gamma^2 \right) + \overline{W}_{ij},\ i \in N \setminus U_k, \tag{14}$$

$$H_{ij}^{(k)} \geq x_i^{(k)} - x_j^{(k)},\ H_{ij}^{(k)} \geq x_j^{(k)} - x_i^{(k)},\ i \in N \setminus U_k, \tag{15}$$

$$\underline{a}_i^{(k)} \leq x_i^{(k)} \leq \overline{a}_i^{(k)},\ k = 1, ..., m,\ i = 1, ..., n. \tag{16}$$

The problem is solved for every $j = 1, ..., n$ and for every $k = 1, ..., T$.

3.5 An algorithm

A general algorithm can be written as follows.

Step 1. $\mathcal{E}(\mathcal{M}_v)$ is the set of extreme points $\alpha^{(1)}, ..., \alpha^{(T)}$.

Step 2. We select the k-th extreme point $\alpha^{(k)}$ from $\mathcal{E}(\mathcal{M}_v)$.

Step 3. For the given $j \in \{1, ..., n\}$ and the selected $k \in \{1, ..., T\}$, we solve linear programming problem (11)-(16).

Step 3. For the given $j \in \{1, ..., n\}$, we select a single value k_j^* such that objective function (11) achieves its maximum, i.e., $k_j^* \leftarrow \arg_k \max O(j,k)$.

Step 5. We select a single value j^* such that objective function (11) achieves its minimum, i.e., $j^* \leftarrow \arg_j \min O(j, k_j^*)$. As a result, we get an optimal vector $(\mathbf{x}_1^*, ..., \mathbf{x}_n^*)$ and an optimal extreme point $\alpha^{(k)}$.

Step 6. This step can be realized in two different ways. The first way is to get the solution φ of problem (4) which can be regarded as the dual one for problem (11)-(16). This step can be carried out by means of the well-known procedures implemented, for example, in the package "linprog" in R-project. The second way is just to solve the problem (2)-(3) by substituting the optimal vector $(\mathbf{x}_1^*, ..., \mathbf{x}_n^*)$ into the objective function (2) and constraints (3).

3.6 Decision strategies for testing

The next question is how to decide whether a testing observation is abnormal or not when it is interval-valued.

The following three decision strategies can be used in order to decide whether the interval-valued observation is normal or abnormal:

1. Classification by using centers of intervals (CC). According to this strategy, every hyper-rectangle is replaced by its center. This is the simplest strategy.

2. Classification by using all vertices of hyper-rectangles produced by intervals and their centers (CA). Here the vertex is a point belonging to one of the bounds of intervals.

3. Classification by using half of points of every hyper-rectangle (CH). According to this strategy, every interval is divided into $k-1$ subintervals (k points for every feature). So, the interval-valued example is viewed as a grid.

These strategies are also used in order to estimate the accuracy of classification for models with interval-valued data

4 Numerical experiments

The model proposed in this paper is illustrated via several examples; all computations have been performed using the programming language Java. We investigate the performance of the proposed model and compare it with C-B model by considering the accuracy (ACC), which is the proportion of correctly classified cases on a sample of data and is often used to quantify the predictive performance of classification methods. By applying C-B model, we take centers of hyper-rectangles produced by intervals from sample data.

It should be noted that all strategies replace the interval-valued training example by a single point or by a set of points. The indicator decision function of correct classification is of the form:

$$C(x) = I(y \cdot f(\mathbf{x}) \geq 0),$$

where $\mathbf{x}$ is a testing example, $f(\mathbf{x})$ is a value of the separating function at point $\mathbf{x}$, $y \in \{-1; 1\}$ is the class label, I is the indicator function.

It should be noted that the labels y are unknown for the classifier. However, in order to evaluate it, testing examples are divided into two classes whose labels are -1 for abnormal examples and 1 for other examples. A class of a point corresponds to the class of the corresponding interval-valued example producing this point with respect to the accepted testing strategy.

Let us formally define the decision functions for every strategy:

1. According to the first strategy, an example is supposed to be correctly classified if the point corresponding to its hyper-rectangle center $\mathbf{x}_i^*$ is correctly classified, i.e.,
$$S(\mathbf{x}_i) = C(\mathbf{x}_i^*).$$

2. According to the second strategy, an example is supposed to be correctly classified if all vertices V_i of the corresponding hyper-rectangle as well as its center $\mathbf{x}_i^*$ are correctly classified, i.e.,
$$S(\mathbf{x}_i) = I\left(1 + |V_i| = \sum_{\mathbf{x} \in V_i \cup \mathbf{x}_i^*} C(\mathbf{x})\right).$$

3. According to the third strategy, an example is supposed to be correctly classified if at least a half of points belonging to the predefined grid G_i of the corresponding hyper-rectangle are correctly classified, i.e.,
$$S(\mathbf{x}_i) = \begin{cases} 1, & \sum_{\mathbf{x} \in G_i} I(C(\mathbf{x}) = 1) \geq \sum_{\mathbf{x} \in G_i} I(C(\mathbf{x}) = 0), \\ 0, & \text{otherwise.} \end{cases}$$

The classification accuracy measure for strategies 1-3 can be computed as follows:
$$ACC = \frac{1}{|X|} \sum_{i=1}^{|X|} S(\mathbf{x}_i),$$
where X is the testing set, $|X|$ is the number of elements in the set X, i.e., the number of testing examples, S is the binary decision function for every strategy such that $S(\mathbf{x}_i) = 1$ if the i-th example is correctly classified, and $S(\mathbf{x}_i) = 0$ otherwise.

We denote the accuracy measure of the proposed model as ACC_S^{Int}, the accuracy measure of the standard C-B model with using the center points as ACC_S^{CB}, where S corresponds to one of the above decision strategies.

4.1 Synthetic data

First of all, we consider the performance of the proposed model with synthetic data having two features $x^{(1)}$ and $x^{(2)}$. The training set consisting of N examples from two subsets is generated in accordance with the following rules. All experiments use the triangular kernel with the kernel parameter γ. Different values for the parameter γ have been tested choosing those leading to the best results.

Generation of normal observations. A set of $N_1 = (1 - \varepsilon)N$ normal observations are generated. Here ε is the portion of abnormal observations.

Step 1. The center of an example $\mathbf{x}_i^*$ denoted as (x_1^*, x_2^*) is generated with respect to the normal probability distribution with expectations $(m_1^{(1)}, m_2^{(1)})$ and with standard deviations $(\sigma_1^{(1)}, \sigma_2^{(1)})$.

Step 2. For every pair (x_1^*, x_2^*), we generate interval-valued pair $(x_1^* - \Delta_1, x_1^* + \Delta_1;\ x_2^* - \Delta_2, x_2^* + \Delta_2)$. The shifts (Δ_1, Δ_2) are generated with respect to the uniform probability distribution with a predefined largest shift M_Δ.

Additionally, we introduce the portion β of one-dimensional intervals, which means how many observations have one point-valued feature, i.e., there holds $\Delta_i = 0$.

Generation of abnormal observations. A set of $N_2 = \varepsilon N$ abnormal observations are generated. Two approaches are used for do it.

1. Normal and abnormal observations are concentrated around two centers defined by different mean values $m_1^{(1)}, m_2^{(1)}$ and $m_1^{(2)}, m_2^{(2)}$, respectively. The observations are governed by the normal probability distributions with identical variances.

2. Normal and abnormal observations are concentrated around one center defined by mean values $m_1^{(1)}$, $m_2^{(1)}$, but with different variances or standard deviations $(\sigma_1^{(1)}, \sigma_2^{(1)})$ and $(r\sigma_1^{(2)}, r\sigma_2^{(2)})$, respectively. The observations are governed by the normal probability distributions with identical mean values. Here r is a multiplier used for generating abnormal observations. A part of abnormal observations which are located close to the center is removed.

We use various values of the kernel parameter γ and the parameter ν in a predefined grid. However, we show only the values that provide the best classification accuracy. Approaches for generation of testing sets are similar to generation of training sets.

First, we study how the classification accuracy depends on values of the largest shift M_Δ, i.e. on the size of intervals. By applying the first approach for generating the training sets, we use the following parameters:

$$N = 50,\ \varepsilon = 0.2,$$
$$(m_1^{(1)}, m_2^{(1)}) = (0, 0),\ (m_1^{(2)}, m_2^{(2)}) = (4, 4),$$
$$(\sigma_1^{(1)}, \sigma_2^{(1)}) = (1, 1),\quad M_\Delta = 0.01, ..., 7,$$
$$\beta = 0.1,\ \gamma = 4,\ \nu = 0.02.$$

Results of testing of the proposed model and the standard C-B model for all decision strategies are shown in Table 1. It can be seen from the table that the proposed model outperforms the C-B model for large values of M_Δ, i.e., for the case of large intervals of training data.

It is interesting also to study how the difference of the accuracy measures depends on M_Δ. The relative difference for strategy S is defined as follows:

$$d = \frac{ACC_S^{\mathrm{Int}} - ACC_S^{\mathrm{CB}}}{(ACC_S^{\mathrm{Int}} + ACC_S^{\mathrm{CB}})/2}.$$

Fig. 2 illustrates the dependence of relative differences between the accuracy measures of the proposed model and the C-B model on values of M_Δ for the

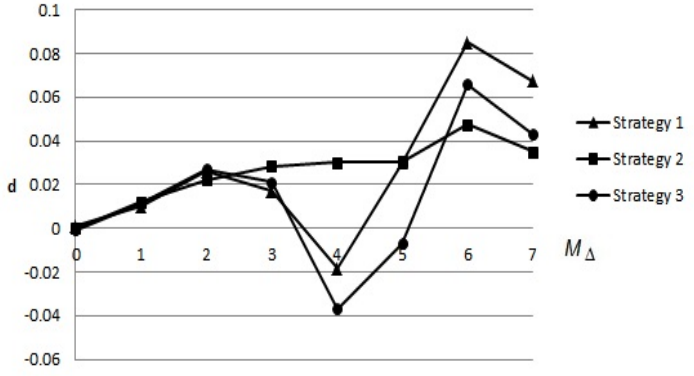

Figure 2: Relative differences d as functions of M_Δ

considered strategies. It can be seen from Fig. 2 that the performance of the proposed model is improved with increase of intervals. It is interesting to note that the C-B model outperforms the proposed model by $M_\Delta = 4$ for all strategies except for the second one. This behavior of the accuracy measures may be associated with the difference between expectations of generated normal and abnormal points.

Second, we study how the classification accuracy depends on the standard deviation σ of features. For generating the training sets, we use the first approach and the following parameters:

$$N = 50,\ \varepsilon = 0.2,$$
$$(m_1^{(1)}, m_2^{(1)}) = (0,0),\ (m_1^{(2)}, m_2^{(2)}) = (8,8),$$
$$(\sigma_1^{(1)}, \sigma_2^{(1)}) = (\sigma, \sigma),\quad M_\Delta = 2,\quad \beta = 0.1,$$
$$\gamma = 4,\ \nu = 0.02.$$

The experimental results are given in Table 2. One can see from Table 2 that the accuracy measures of the proposed and standard C-B models converge to close values with increase of the standard deviation σ. At the same time, the proposed model provides better results when the standard deviation is rather small.

Fig. 3 illustrates the dependence of relative differences between the accuracy measures of the proposed model and the C-B model on the values of the standard deviation for the considered strategies.

Third, we study how the classification accuracy depends on the location of normal and abnormal observations. By using the first approach for generating interval-valued training data (normal and abnormal observations are concentrated around two centers), we investigate the classifiers by the following

Table 1: The classification accuracy of proposed and C-B models for different strategies and different values of the shift M_Δ

	Strategy 1		Strategy 2		Strategy 3	
M_Δ	ACC_{CC}^{CB}	ACC_{CC}^{Int}	ACC_{CA}^{CB}	ACC_{CA}^{Int}	ACC_{CH}^{CB}	ACC_{CH}^{Int}
0.01	0.7823	0.7832	0.7831	0.7830	0.7834	0.7829
0.1	0.7877	0.7864	0.7902	0.7886	0.7878	0.7864
0.5	0.7779	0.7771	0.7897	0.7899	0.7766	0.7758
1.0	0.7855	0.7936	0.7978	0.8075	0.7833	0.7923
1.5	0.7842	0.8142	0.7983	0.8161	0.7805	0.8128
2.0	0.7828	0.8036	0.7973	0.8151	0.7781	0.7994
2.5	0.7762	0.7954	0.7967	0.8159	0.7676	0.7903
3.0	0.7869	0.8005	0.7995	0.8226	0.7730	0.7896
4.0	0.7822	0.7680	0.7975	0.8219	0.7506	0.7236
5.0	0.7808	0.8044	0.7973	0.8221	0.7184	0.7135
6.0	0.7861	0.8561	0.7967	0.8356	0.6698	0.7155
7.0	0.7902	0.8453	0.7977	0.8262	0.5984	0.6248

Table 2: The classification accuracy of the proposed and C-B models for different strategies and different values of the the standard deviation of features

	Strategy 1		Strategy 2		Strategy 3	
σ	ACC_{CC}^{CB}	ACC_{CC}^{Int}	ACC_{CA}^{CB}	ACC_{CA}^{Int}	ACC_{CH}^{CB}	ACC_{CH}^{Int}
0.25	0.7603	0.8542	0.8079	0.9470	0.4705	0.7415
0.5	0.7548	0.8597	0.8029	0.9114	0.6938	0.8638
0.75	0.7598	0.8578	0.8031	0.8861	0.7589	0.8790
1.0	0.7658	0.8419	0.8027	0.8550	0.7697	0.8587
1.5	0.7563	0.7875	0.7936	0.8169	0.7565	0.7915
2.0	0.7522	0.7651	0.7889	0.8000	0.7520	0.7661
2.5	0.7313	0.7309	0.7756	0.7780	0.7318	0.7317
3.0	0.7102	0.7136	0.7645	0.7677	0.7122	0.7158

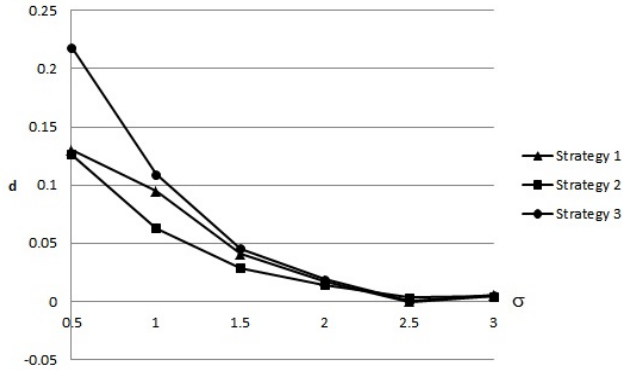

Figure 3: Relative differences d as functions of the standard deviation

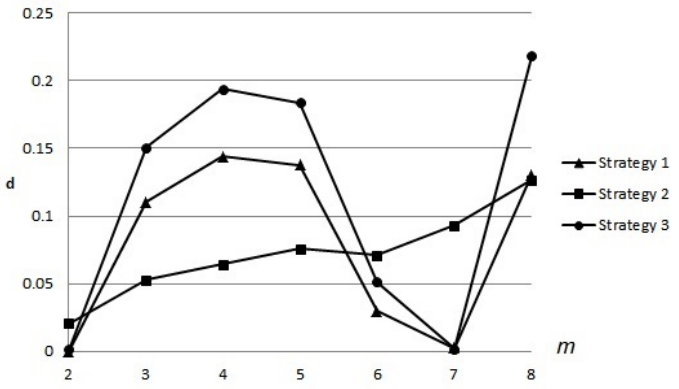

Figure 4: Relative differences d as functions of the distance between expectations of normal and abnormal data

parameters:

$$N = 50,\ \varepsilon = 0.2,$$
$$(m_1^{(1)}, m_2^{(1)}) = (0,0),\ (m_1^{(2)}, m_2^{(2)}) = (m,m),$$
$$(\sigma_1^{(1)}, \sigma_2^{(1)}) = (0.5, 0.5),\quad M_\Delta = 2,\quad \beta = 0.1,$$
$$\gamma = 4,\ \nu = 0.02.$$

One can see that the mean values of abnormal observations are taken as (m, m) in order to consider the dependence of the classification accuracy measures on m. By changing the mean values $(m_1^{(2)}, m_2^{(2)})$, we change the distance between normally distributed examples and abnormal observations. Table 3 shows how the accuracy measures depend on this distance for different decision strategies. It can be seen from the table that the proposed model outperforms the C-B model for large values of m.

Relative differences between the accuracy measures as functions of the distance between expectations of generated normal and abnormal observations are depicted in Fig. 4. It is interesting to see from Fig. 4 that the difference between accuracy measures of the proposed model and the C-B model increases for some strategies before $m = 4$ and then decreases after this distance value. It can be explained as follows. When m is rather small, i.e., normal and abnormal observations are very close to each other, the proposed model as well as the C-B model are identical and show similar bad results. This can be seen from Table 3 for $m = 2$. After increasing m, one could expect that the C-B model with using the center points of intervals will outperform the proposed model. However, we can see that the width of intervals determined by $M_\Delta = 2$ is comparable with m. This leads to outperforming of the proposed model for all strategies. When m is rather large in comparison with given M_Δ, the difference between models decreases and may be very small.

Table 3: The classification accuracy of the proposed and C-B models for different strategies and different mean values of abnormal observations by $(\sigma_1^{(1)}, \sigma_2^{(1)}) = (0.5, 0.5)$

	Strategy 1		Strategy 2		Strategy 3	
m	ACC_{CC}^{CB}	ACC_{CC}^{Int}	ACC_{CA}^{CB}	ACC_{CA}^{Int}	ACC_{CH}^{CB}	ACC_{CH}^{Int}
2.0	0.7842	0.7842	0.7982	0.8149	0.7558	0.7569
2.5	0.7800	0.7869	0.8146	0.7999	0.8333	0.7617
3.0	0.7834	0.8751	0.7988	0.8422	0.7560	0.8790
3.5	0.7890	0.8985	0.7989	0.8569	0.7664	0.9088
4.0	0.7818	0.9029	0.7987	0.8519	0.7542	0.9159
4.5	0.7959	0.9055	0.8011	0.8695	0.7750	0.9174
5.0	0.7868	0.9032	0.7997	0.8625	0.7626	0.9170
6.0	0.7879	0.8115	0.8000	0.8589	0.7628	0.8029
7.0	0.7964	0.7982	0.8025	0.8808	0.7781	0.7796
8.0	0.7545	0.8600	0.8025	0.9112	0.6936	0.8635

Similar results can be obtained by increasing the standard deviation of features $(\sigma_1^{(1)}, \sigma_2^{(1)}) = (1, 1)$. They are shown in Table 4. It can be seen from the table that the relative quality of the proposed model depends on the strategy. In particular, the second strategy provides the best results in comparison with other strategies.

It has been pointed out in Section 1 that there are other one-class classification models, for example, the well-known model proposed by Tax and Duin [44, 45] (the T-D model). Therefore, we also use this model for its comparison with the proposed model by interval-valued data. Table 5 illustrates the difference between accuracy measures of the T-D model (ACC_{CC}^{TD}) and the proposed model (ACC_{CC}^{Int}) for the first strategy by the standard deviation of features $(\sigma_1^{(1)}, \sigma_2^{(1)}) = (0.5, 0.5)$ and by different mean values of abnormal observations. One can observe that the T-D model provides better results in comparison with the C-B model only for some values of m (see Table 3 providing similar numerical results with the C-B model for comparison). However, the character of the relationship between this model and the proposed model is the same.

The same numerical experiments can be provided by using the second approach for generating abnormal observations, namely, when normal and abnormal observations are concentrated around one center, but with different variances. We use the following parameters for generating training data:

$$\begin{gathered} N = 50,\ \varepsilon = 0.2, \\ (m_1^{(1)}, m_2^{(1)}) = (0, 0),\ (\sigma_1^{(1)}, \sigma_2^{(1)}) = (0.5, 0.5), \\ M_\Delta = 0.01, ..., 7, \\ \beta = 0.1,\ \gamma = 4,\ \nu = 0.02,\ r = 3. \end{gathered}$$

The corresponding experimental results are given in Table 6. Relative dif-

Table 4: The classification accuracy of the proposed and C-B models for different strategies and different mean values of abnormal observations by $(\sigma_1^{(1)}, \sigma_2^{(1)}) = (1, 1)$

	Strategy 1		Strategy 2		Strategy 3	
m	ACC_{CC}^{CB}	ACC_{CC}^{Int}	ACC_{CA}^{CB}	ACC_{CA}^{Int}	ACC_{CH}^{CB}	ACC_{CH}^{Int}
2.0	0.7801	0.7651	0.7973	0.8007	0.7743	0.7555
2.5	0.7816	0.7862	0.7986	0.8094	0.7747	0.7800
3.0	0.7754	0.7750	0.7956	0.8009	0.7682	0.7701
3.5	0.7818	0.7936	0.7974	0.8101	0.7767	0.7899
4.0	0.7859	0.7931	0.7998	0.8098	0.7808	0.7890
4.5	0.7859	0.8034	0.7999	0.8151	0.7811	0.8014
5.0	0.7844	0.8153	0.7999	0.8285	0.7786	0.8148
6.0	0.7873	0.7840	0.8019	0.8250	0.7876	0.7897
7.0	0.7715	0.8198	0.8016	0.8602	0.7783	0.8401
8.0	0.7651	0.8408	0.8028	0.8553	0.7694	0.8585

Table 5: The classification accuracy of the proposed and T-D models for the first strategy and different mean values of abnormal observations by $(\sigma_1^{(1)}, \sigma_2^{(1)}) = (0.5, 0.5)$

m	ACC_{CC}^{TD}	ACC_{CC}^{Int}
2.0	0.7785	0.7842
3.0	0.7801	0.8751
4.0	0.8151	0.9029
5.0	0.8401	0.9032
6.0	0.8032	0.8115
7.0	0.7981	0.7982
8.0	0.8322	0.8600

Table 6: The classification accuracy of the proposed and C-B models for different strategies and different values of the shift M_Δ

	Strategy 1		Strategy 2		Strategy 3	
M_Δ	ACC_{CC}^{CB}	ACC_{CC}^{Int}	ACC_{CA}^{CB}	ACC_{CA}^{Int}	ACC_{CH}^{CB}	ACC_{CH}^{Int}
0.01	0.8957	0.8956	0.8942	0.8951	0.8960	0.8959
0.1	0.8959	0.9008	0.8874	0.8926	0.8959	0.9005
0.5	0.8968	0.9110	0.8592	0.8770	0.8965	0.9109
1.0	0.8930	0.9057	0.8358	0.8543	0.8952	0.9063
1.5	0.9005	0.8942	0.8289	0.8443	0.9076	0.8928
2.0	0.8948	0.8839	0.8238	0.8408	0.9084	0.8709

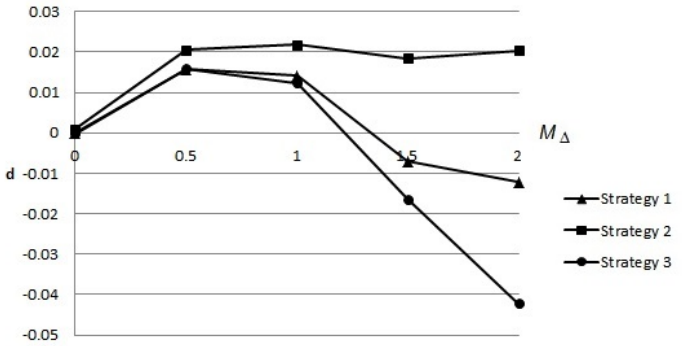

Figure 5: Relative differences d as functions of M_Δ for the second generation approach

ferences between the accuracy measures as functions of M_Δ are depicted in Fig. 5.

The next experiment aims to investigate how the parameter r impacts on the accuracy measures of the models. We again use the second approach for generating abnormal observations with parameters:

$$N = 50,\ \varepsilon = 0.2,$$
$$(m_1^{(1)}, m_2^{(1)}) = (0,0),\ (\sigma_1^{(1)}, \sigma_2^{(1)}) = (0.5, 0.5),$$
$$M_\Delta = 1,\ \beta = 0.1,\ \gamma = 4,\ \nu = 0.02,$$
$$r = 2, ..., 10.$$

The corresponding experimental results are given in Table 7. Relative differences between the accuracy measures as functions of r are depicted in Fig. 6.

An example of separating functions computed by means of the proposed model (thick curve) and the C-B model with using centers of hyper-rectangles (dashed curve) is shown in Fig. 7. The abnormal observations are generated

Table 7: The classification accuracy of the proposed and C-B models for different strategies and different values of r

	Strategy 1		Strategy 2		Strategy 3	
r	ACC_{CC}^{CB}	ACC_{CC}^{Int}	ACC_{CA}^{CB}	ACC_{CA}^{Int}	ACC_{CH}^{CB}	ACC_{CH}^{Int}
2.0	0.8790	0.8722	0.8294	0.8370	0.8822	0.8713
2.5	0.8907	0.8926	0.8343	0.8488	0.8939	0.8941
3.0	0.9011	0.9132	0.8406	0.8595	0.9031	0.9140
3.5	0.8989	0.9110	0.8397	0.8580	0.8999	0.9122
4.0	0.8979	0.9152	0.8392	0.8607	0.8998	0.9167
5.0	0.9003	0.9235	0.8436	0.8668	0.9014	0.9247
10.0	0.9022	0.9272	0.8556	0.8788	0.9021	0.9272

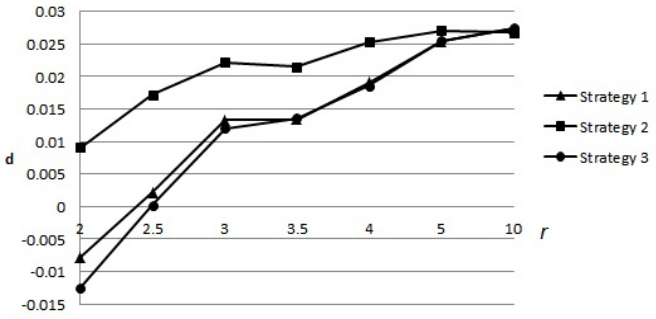

Figure 6: Relative differences d as functions of r for the second generation approach

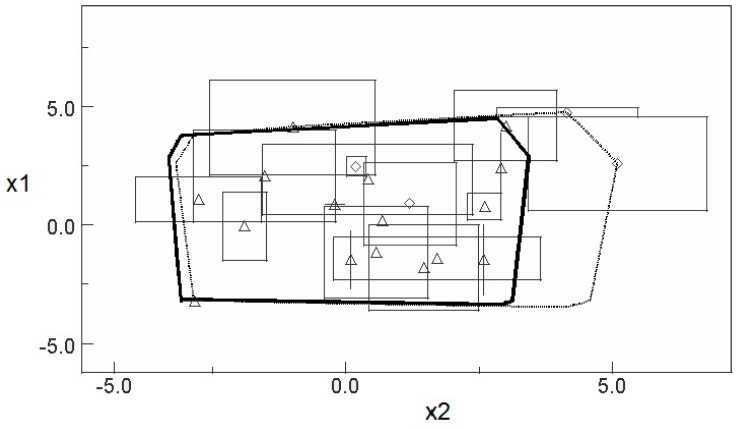

Figure 7: An example of separating function computed by different models for the first generation approach

by applying the first approach, i.e., normal and abnormal observations are concentrated around two centers. Centers of hyper-rectangles corresponding to normal and abnormal observations are depicted by small triangles and diamonds, respectively. The following parameters are used for generating random interval-valued observations: $N = 20$, $\varepsilon = 0.2$, $(m_1^{(1)}, m_2^{(1)}) = (0, 0)$, $(m_1^{(2)}, m_2^{(2)}) = (4, 4)$, $(\sigma_1^{(1)}, \sigma_2^{(1)}) = (2, 2)$, $M_\Delta = 2$, $\beta = 0.1$, $\gamma = 4$, $\nu = 0.02$.

Another example of similar separating functions is shown in Fig. 8. The abnormal observations in this case are generated by applying the second approach, i.e., normal and abnormal observations are concentrated around one center, but with different variances. The following parameters are used for generating interval-valued observations: $N = 30$, $\varepsilon = 0.2$, $(m_1^{(1)}, m_2^{(1)}) = (0, 0)$, $(\sigma_1^{(1)}, \sigma_2^{(1)}) = (0.5, 0.5)$, $M_\Delta = 1$, $\beta = 0.1$, $\gamma = 4$, $\nu = 0.02$, $r = 2.5$.

4.2 Real data

The proposed model has been evaluated and investigated by the following publicly available data sets: Indian Liver Patient, Iris, Vertebral Column, Seeds, Glass Identification. All data sets are from the UCI Machine Learning Repository [20]. The following is a brief introduction about these data sets, while more detailed information can be found from, respectively, the data resources.

Indian Liver Patient Data set (ILPD) contains 416 liver patient records and 167 non-liver patient records characterized by 10 features. Liver patients are viewed as normal data, non-liver patients are abnormal.

Iris data set contains 3 classes (Iris Setosa, Iris Versicolour, Iris Virginica) of 50 instances each. The number of features is 4 (sepal length, sepal width, petal length, petal width). It is supposed that data points from the Iris Setosa class are abnormal, i.e., the number of abnormal examples is 30.

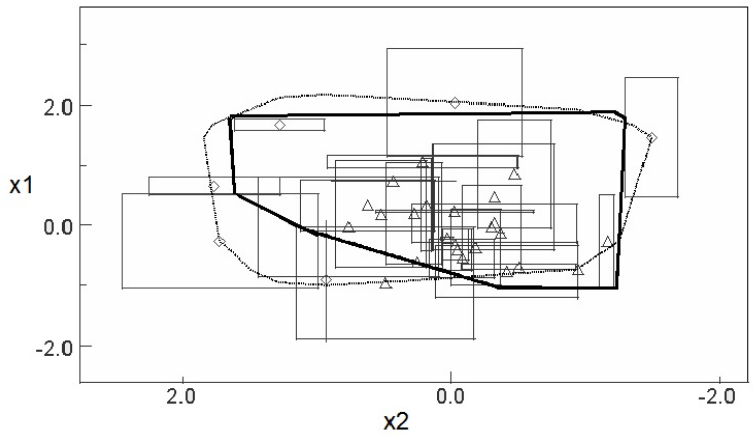

Figure 8: An example of separating functions computed by different models for the second generation approach

Vertebral Column data set contains 2 classes of patients. The classes consist of 100 patients considered as normal and 210 patients considered as abnormal. Each patient is represented in the data set by six biomechanical attributes derived from the shape and orientation of the pelvis and lumbar spine.

Seeds data set consists of three different varieties of wheat: Kama, Rosa and Canadian, 70 elements each. Elements are characterized by 7 features. For using the set in one-class classification, elements of the class Canadian are supposed to be abnormal. Other elements are normal.

Glass Identification data set contains 6 types of glass and totally 214 examples having 10 features. Elements of the first and second classes are viewed as normal. Other elements are abnormal.

The following algorithm is used for numerical experiments with real data:

Step 1. All examples of the training set are divided into training and testing subsets such that the training set contains 40 examples.

Step 2. For all examples and for every feature, the intervals are randomly generated similarly to generation of intervals for synthetic data. The center of the interval-valued example is the original value of every feature in real data. The largest shift M_Δ is equal to the sample standard deviation computed by using the total training set multiplied by the introduced interval deviation coefficient (IDC).

Step 3. Separating functions are computed for the standard C-B model and for the proposed model on the basis of training subset.

Step 4. The classification accuracy for every decision strategy is determined on the basis of the testing subsets in accordance with every strategy.

It should be noted that the accuracy measures are computed as average values by means of the random selection of training and testing subsets from data sets many times.

Table 8: Optimal values of parameters for real data sets

	γ	IDC
ILPD	16	2
Vertebral	4	5
Seeds	2	10
Iris	2	5
Glass	2	5

Table 9: Accuracy measures for real data sets

	Strategy 1		Strategy 2		Strategy 3	
	ACC_{CC}^{CB}	ACC_{CC}^{Int}	ACC_{CA}^{CB}	ACC_{CA}^{Int}	ACC_{CH}^{CB}	ACC_{CH}^{Int}
ILPD	0.6066	0.7220	0.6070	0.7102	0.6005	0.7124
Vertebral	0.6750	0.6945	0.6750	0.7140	0.6750	0.6815
Seeds	0.6397	0.8065	0.6459	0.8462	0.6460	0.8400
Iris	0.5030	0.7755	0.4886	0.7491	0.4885	0.8450
Glass	0.4168	0.8014	0.4068	0.8126	0.4161	0.8909

The kernel parameter γ for every data set is selected separately. The optimal values of γ and the IDC providing the largest classification accuracy are shown in Table 8. The accuracy measures are shown in Table 9.

5 Conclusion

A new OCC model dealing with interval-valued training data has been proposed in the paper. Many experiments have shown that the model outperforms the standard methods especially when mainly large intervals of training data are available. The proposed model comes to a finite set of simple linear programming problems whose solution does not meet difficulties. Another advantage of the proposed model is that we can find "optimal" points of intervals corresponding to the robust or maximin decision strategy.

At the same time, a bottle neck of the proposed model is its complexity by computing the extreme points $\alpha^{(1)}, ..., \alpha^{(T)}$ when v is rather large. It is obvious that the value T may be very large. If we compare this model with the similar model developed by Utkin et al. [47] which is based on enumerating vertices of polytopes produced by the intervals $\mathbf{A}_1, ..., \mathbf{A}_n$, then its use in contrast to the model [47] is efficient when the number of features is rather large and the number of examples is small. This is an important condition of the model usage. Indeed, one can see that the number of extreme points $\alpha^{(1)}, ..., \alpha^{(T)}$ does not depend on the number of features. By returning to the model in [47], it strictly depends on numbers of features and examples.

One of the ideas allowing to come to simple linear problems is the use of the triangular kernel instead of the Gaussian one. However, we have to point out that the obtained optimization problems have many constraints due to replace-

ments of absolute values which take place in the triangular kernel. Another idea to avoid the absolute values is to apply the so called Epanechnikov kernel which can be regarded as a quadratic approximation. This idea leads to quadratically constrained linear programming problems. Efficient algorithms for solving these problems are directions for further work. Another idea is to extend the proposed model and the use of triangular kernel on the binary classification problem. This is also a direction for further research.

Acknowledgement

The reported study was partially supported by RFBR, research project No. 15-01-01414-a. The authors would like to express their appreciation to the anonymous referees whose very valuable comments have improved the paper.

References

[1] C. Angulo, D. Anguita, L. Gonzalez-Abril, and J.A. Ortega. Support vector machines for interval discriminant analysis. *Neurocomputing*, 71(7-9):1220 – 1229, 2008.

[2] A.M. Bartkowiak. Anomaly, novelty, one-class classification: A comprehensive introduction. *International Journal of Computer Information Systems and Industrial Management Applications*, 3:61–71, 2011.

[3] O. Beaumont. Solving interval linear systems with linear programming techniques. *Linear Algebra and Its Applications*, 281:293–309, 1998.

[4] S. Bhadra, J.S. Nath, A. Ben-Tal, and C. Bhattacharyya. Interval data classification under partial information: A chance-constraint approach. In T. Theeramunkong, B. Kijsirikul, N. Cercone, and T.-B. Ho, editors, *Advances in Knowledge Discovery and Data Mining*, volume 5476 of *Lecture Notes in Computer Science*, pages 208–219. Springer, Berlin Heidelberg, 2009.

[5] M. Bicego and M.A.T. Figueiredo. Soft clustering using weighted one-class support vector machines. *Pattern Recognition*, 42(1):27–32, 2009.

[6] C. Campbell. Kernel methods: a survey of current techniques. *Neurocomputing*, 48(1-4):63–84, 2002.

[7] C. Campbell and K.P. Bennett. A linear programming approach to novelty detection. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 395–401. MIT Press, 2001.

[8] E. Carrizosa, J. Gordillo, and F. Plastria. Classification problems with imprecise data through separating hyperplanes. Technical Report MOSI/33, MOSI Department, Vrije Universiteit Brussel, September 2007.

[9] E. Carrizosa, J. Gordillo, and F. Plastria. Support vector regression for imprecise data. Technical Report MOSI/35, MOSI Department, Vrije Universiteit Brussel, October 2007.

[10] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. Technical Report TR 07-017, University of Minnesota, Minneapolis, MN, USA, 2007.

[11] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

[12] M. Chavent. A hausdorff distance between hyper-rectangles for clustering interval data. In *Classification, Clustering, and Data Mining Applications*, pages 333–339. Springer Berlin Heidelberg, 2004.

[13] M. Chavent, F. de A.T. de Carvalho, Y. Lechevallier, and R. Verde. New clustering methods for interval data. *Computational statistics*, 21(2):211–229, 2006.

[14] V. Cherkassky and F.M. Mulier. *Learning from Data: Concepts, Theory, and Methods*. Wiley-IEEE Press, UK, 2007.

[15] R.M.C.R. de Souza and F. de A.T. de Carvalho. Clustering of interval data based on city-block distances. *Pattern Recognition Letters*, 25:353–365, 2004.

[16] Thanh-Nghi Do and F. Poulet. Kernel methods and visualization for interval data mining. In *Internaional Symposium on Applied Stochastic Models and Data Analysis, ASMDA*, volume 5, pages 345–355, 2005.

[17] M. Ferecatu, N. Boujemaa, and M. Crucianu. Semantic interactive image retrieval combining visual and conceptual content description. *ACM Multimedia Systems Journal*, 13(5-6):309–322, 2008.

[18] F. Fleuret and H. Sahbi. Scale-invariance of support vector machines based on the triangular kernel. In *3rd International Workshop on Statistical and Computational Theories of Vision*, 2003.

[19] Y. Forghani and H.S. Yazdi. Robust support vector machine-trained fuzzy system. *Neural Networks*, 50:154–165, 2014.

[20] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[21] L.E. Ghaoui, G.R.G. Lanckriet, and G. Natsoulis. Robust classification with interval data. Technical Report Report No. UCB/CSD-03-1279, University of California, Berkeley, California 94720, 2003.

[22] P.-Y. Hao. Interval regression analysis using support vector networks. *Fuzzy Sets and Systems*, 60:2466–2485, 2009.

[23] V.J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[24] H. Ishibuchi, H. Tanaka, and N. Fukuoka. Discriminant analysis of multidimensional interval data and its application to chemical sensing. *International Journal of General Systems*, 16(4):311–329, 1990.

[25] S.S. Khan and M.G. Madden. A survey of recent trends in one class classification. In L. Coyle and J. Freyne, editors, *Artificial Intelligence and Cognitive Science*, volume 6206 of *Lecture Notes in Computer Science*, pages 188–197. Springer Berlin / Heidelberg, 2010.

[26] J.T. Kwok, I.W.-H. Tsang, and J.M. Zurada. A class of single-class minimax probability machines for novelty detection. *IEEE Transactions on Neural Networks*, 18(3):778–785, 2007.

[27] Y. Li. Selecting training points for one-class support vector machines. *Pattern Recognition Letters*, 32(11):1517–1522, 2011.

[28] L.M. Manevitz and M. Yousef. One-class SVMs for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

[29] M. Markou and S. Singh. Novelty detection: a review—part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003.

[30] A. Musdholifah and S.Z.M. Hashim. Cluster analysis on high-dimensional data: A comparison of density-based clustering algorithms. *Australian Journal of Basic and Applied Sciences, 7(2): 380-389, 2013*, 7(2):380–389, 2013.

[31] E.A. Lima Neto and F.A.T. de Carvalho. Centre and range method to fitting a linear regression model on symbolic interval data. *Computational Statistics and Data Analysis*, 52:1500–1515, 2008.

[32] P. Nivlet, F. Fournier, and J.-J. Royer. Interval discriminant analysis: An efficient method to integrate errors in supervised pattern recognition. In *Second International Symposium on Imprecise Probabilities and Their Applications*, pages 284–292, Ithaca, NY, USA, 2001.

[33] W. Pedrycz, B.J. Park, and S.K. Oh. The design of granular classifiers: A study in the synergy of interval calculus and fuzzy sets in pattern recognition. *Pattern Recognition*, 41(12):3720–3735, 2008.

[34] K. Pelckmans, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684 – 692, 2005.

[35] C.P. Robert. *The Bayesian Choice*. Springer, New York, 1994.

[36] Hichem Sahbi. Kernel PCA for similarity invariant shape recognition. *Neurocomputing*, 70(16-18):3034–3045, 2007.

[37] B. Scholkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[38] B. Scholkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Cambridge, Massachusetts, 2002.

[39] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 526–532. 2000.

[40] G. Schollmeyer and T. Augustin. On sharp identification regions for regression under interval data. In F. Cozman, T. Denœux, S. Destercke, and T. Seidenfeld, editors, *ISIPTA'13: Proceedings of the Eighth International Symposium on Imprecise Probability: Theories and Applications*, pages 285–294, Compiègne, 2013. SIPTA.

[41] A. Silva and P. Brito. Linear discriminant analysis for interval data. *Computational Statistics*, 21:289–308, 2006.

[42] A.J. Smola and B. Scholkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.

[43] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6:211–232, December 2005.

[44] D.M.J. Tax and R.P.W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[45] D.M.J. Tax and R.P.W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11):1191–1199, 1999.

[46] L.V. Utkin. A framework for imprecise robust one-class classification models. *International Journal of Machine Learning and Cybernetics*, 5(3):379–393, 2014.

[47] L.V. Utkin, Y.A. Zhuk, and A.I. Chekh. A robust one-class classification model with interval-valued data based on belief functions and minimax strategy. In P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 8556 of *Lecture Notes in Computer Science*, pages 107–118. Springer, 2014.

[48] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[49] J. Wang, H. Lu, K.N. Plataniotis, and J. Lu. Gaussian kernel optimization for pattern classification. *Pattern Recognition*, 42(7):1237 – 1247, 2009.

[50] L. Zhang and W.-D. Zhou. 1-norm support vector novelty detection and its sparseness. *Neural Networks*, 48:125–132, 2013.