Neurocomputing 331 (2019) 18–32

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An imprecise extension of SVM-based machine learning models

Lev V. Utkin

Peter the Great St.Petersburg Polytechnic University, Russia

ARTICLE INFO

Article history:
Received 14 July 2016
Revised 15 August 2018
Accepted 18 November 2018
Available online 22 November 2018

Communicated by A. Abraham

Keywords:
Machine learning
Support vector machine
Duality
Classification
Regression
Interval-valued data
Imprecise model

ABSTRACT

A general approach for incorporating imprecise prior knowledge and for robustifying the machine learning SVM-based models is proposed in the paper. The main idea underlying the approach is to use a double duality representation in the framework of the minimax strategy of decision making. This idea allows us to get simple extensions of SVMs including additional constraints for optimization variables (the Lagrange multipliers) formalizing the incorporated imprecise information. The approach is applied to regression, binary classification and one-class classification SVM-based problems. Moreover, it is adopted to set-valued or interval-valued training data. For every problem, numerical examples are provided which illustrate how imprecise information may improve the machine learning algorithm performance.

1. Introduction

Most models of machine learning, including regression and classification models, are based on minimizing the empirical risk instead of the expected risk because the joint probability distribution from which training examples are drawn is usually unknown. Therefore, the empirical risk can be viewed as an approximation of the expected risk under condition that the unknown probability distribution is replaced by an empirical probability distribution constructed on the basis of the training set. Properties of the uniform convergence of the empirical risk to the expected risk over a set of loss functions have been investigated by Vapnik [48] where the Vapnik–Chervonenkis (VC) dimension is introduced in order to provide a bound to the rate of the uniform convergence.

A problem of using the empirical risk minimization is when we have a small training set because, for small samples, it is difficult to guarantee that the empirical risk minimization will also minimize the expected risk. One of the ways to overcome this difficulty is to use robust models which have been exploited in machine learning due to the opportunity to avoid some strong assumptions underlying the standard models. A review of robust models in machine learning was proposed by Xu et al. [53]. There are a lot of published results providing various robust classification and regression models [7,14,33,45] which use an assumption that inputs are subject to an additive noise. Another class of robust models is based on relaxing strong assumptions about a probability distribution of data points (see, for instance, [23]). According to these models, probability distributions of examples or their weights are assumed to be different and may vary within some predefined set of probability distributions.

Another way to deal with small training sets is to incorporate prior knowledge about the problem domain at hand, which may significantly improve the performance of machine learning algorithms in many applications [24,26,29,47]. Prior knowledge may take various forms, ranging from knowledge about the importance of a class, the informativeness of features, the quality of samples to knowledge about the dependency of variables [38].

The precise weights assigned to training examples have been widely used for incorporating the prior information about classes [20,55] and about examples of training sets [5,13,42,54]. One of the important forms of the prior information is a set of weights which can be regarded as a special case of the imprecise information [49]. In order to construct robust classifiers, Mangasarian [29] proposed to incorporate prior knowledge in a form of constraints to a linear program called as a knowledge-based linear program. Fung et al. [12] introduced prior knowledge in a form of multiple polyhedral sets incorporated into a linear SVM classifier. It was clearly shown by Fung et al. [12] that some rules provided by an expert in an applied area can be converted to linear inequalities which produce sets of weights. Li et al. [25] provided a robust conjugate duality theory for convex programming problems in the face of data uncertainty within the framework of robust optimization, and derived a robust conjugate duality theorem for support vector machines.

E-mail addresses: lev.utkin@gmail.com, utkin_lv@spbstu.ru

https://doi.org/10.1016/j.neucom.2018.11.053
0925-2312/

In spite of the large number of publications devoted to incorporating prior knowledge into machine learning algorithms, there is no a unified approach for using imprecise information in standard SVMs which are based on solving a quadratic optimization problem. Therefore, we consider how to incorporate arbitrary information in the form of interval-valued expectations which produce polyhedral sets of weights assigned to training examples into the SVM-based quadratic optimization problem for regression, classification and novelty detection (one-class classification) problems. It is shown in the present paper that constraints for sets of weights can be represented in the same form as constraints to the dual optimization problem (the Lagrangian) in the standard SVM. It should be noted that the proposed approach can be also viewed as a way for robustifying the machine learning algorithms by taking a set of probability distributions around the uniform distribution, which is widely accepted in the empirical expected risk minimization. In other words, the probability $\pi_i = 1/n$ is replaced by the interval $a_i \le \pi_i \le b_i$, where n is the number of examples in a training set, π_i is the probability or the weight assigned to the ith example.

Another advantage of the proposed approach is its extension on set-valued and interval-valued training data. Interval-valued data stem from imperfection of measurement tools or imprecision of expert information, from missing data. Interval training sets may arise in situations of specific processing and representation of point-valued data such as recording monthly interval temperatures in meteorological stations, daily interval stock prices, etc. Another source of interval data is the aggregation of huge data-bases into a reduced number of groups [30]. The problem of interval-valued observations has been studied by many authors [1,3,4,8,9,11,16,31,41,52]. The interest to interval data indicates a large importance of the machine learning algorithms under the interval-valued uncertainty in many applications. It should be noted that the set-valued observations take place in many applications. For example, the temperature sensors in a multi-robot system usually provide the environment temperature from a set of robots. Therefore, the set of temperature measurements in this case should be regarded as a single set-valued training example. It is shown in the present paper that the proposed approach is simply extended on the case of set-valued and interval-valued data.

The main idea underlying the proposed approach is to use a double duality representation in the framework of the minimax strategy of decision making. The first duality is implemented for the set of probability distributions or weights produced by the imprecise prior information about training data. This dual representation is incorporated into the SVM-based primal quadratic optimization problem in the form of an objective function extended by a standard regularization term. Then the dual quadratic optimization problem (the Lagrangian) is derived in a standard way. As a result, bounds for expectations of some functions, which define the prior information, become the bounds for Lagrange multipliers in the second dual optimization problem.

We apply the proposed approach to main machine learning problems which are implemented by using the SVM: regression (SVR), binary classification and one-class classification (OCC SVM) problems. For every problem, we provide numerical examples with synthetic and real data, which illustrate how imprecise information may improve the machine learning algorithm performance. In particular, we illustrate how the proposed models can be applied to the important and well-known applied problems, including the software reliability growth modeling and the structural reliability analysis.

The paper is organized as follows. Section 2 provides a detailed derivation of a SVM-based regression model with imprecise prior knowledge. Interesting special cases are studied in this section. The same derivation for the SVM-based classification problem is studied in Section 3. Two statements of the OCC SVM problems (Scholkopf's OCC SVM model [40] and Tax and Duin's OCC SVM model [43]) are modified in Section 4 in order to take into account imprecise prior knowledge. It is shown in Section 5 how set-valued and interval-valued data can be represented in the framework of the proposed approach. Numerical experiments for every machine learning problem are given in Section 6. Section 7 contains conclusion remarks. Proofs of propositions and the corresponding special cases are given in Appendix.

2. The support vector regression models

2.1. The standard SVR

Let us consider a regression problem of estimating a continuous function h defined on $\mathbb{R}$ and depending on parameters $\mathbf{w} = (w_0, w)$, where $w = (w_1, \ldots, w_m)$. The observed output y for $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^m$ can be represented as $y = h(\mathbf{x}) + \varepsilon$, where ε represents random noise with a zero mean and an unknown variance. If we have a training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ of points $(\mathbf{x}_i, y_i)$, then it is assumed that these points are randomly generated according to the distribution $P(\mathbf{x})$ and $y_i = h(\mathbf{x}_i) + \varepsilon_i$. Here all ε_i are independent and identically distributed (i.i.d.) random variables having the same distribution with ε. The machine learning aims to construct a regression model or an approximation f of the function h that minimizes the expected risk

$$R(f) = \int_{\mathcal{X}\times\mathbb{R}} l(y, f(\mathbf{x}))\mathrm{d}P(\mathbf{x}, y) \tag{1}$$

with respect to the function parameters, where the loss function l may be represented as follows: $l(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$. The function f is often represented as a linear combination of the form $f(\mathbf{x}, \mathbf{w}) = w_0 + \langle w_i, \phi(\mathbf{x})\rangle$. Here $\langle\cdot,\cdot\rangle$ denotes the dot product of two vectors; ϕ is a feature map $\mathbb{R}^m \to G$ such that the data points are mapped into an alternative higher-dimensional feature space G. It is important to note that SVRs as well as SVMs do not deal directly with the feature map ϕ, but they use the dot-product $\langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$ which can be computed by evaluating some simple kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$, for example, the Gaussian kernel that is very popular because it supports many complex models and is rather flexible [51].

The probability distribution $P(\mathbf{x}, y)$ is usually unknown. Therefore, in order to minimize the expected risk $R(f)$, it is replaced with the empirical risk $R_{\mathrm{emp}}(f)$ which can be regarded as the mean of the loss function values at points from the training set S. The empirical risk is

$$R_{\mathrm{emp}}(f) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(\mathbf{x}_i, \mathbf{w})). \tag{2}$$

The standard SVR can be obtained by minimizing (2) over parameters w. In order to restrict the class of admissible solutions and to prevent over-fitting, the objective function (2) is added by a stabilization or regularization term $\Psi[f]$, in particular, $\Psi[f] = \|w\|^2/2 = \langle w, w\rangle/2$. This is the standard Tikhonov regularization term (the most popular penalty or smoothness term) [44]. A detailed analysis of regularization methods can be found also in [39].

The common approach for selecting the loss function $l(y, f(\mathbf{x}, \mathbf{w}))$ in the SVR is the so-called ϵ-insensitive loss function with the parameter ϵ, which is defined as $l(y, f(\mathbf{x}, \mathbf{w})) = \max(0, |y - f(\mathbf{x}, \mathbf{w})| - \epsilon)$. After substituting the ϵ-insensitive loss function into (2) and after adding the regularization term, the regression problem statement can be finally rewritten as the following quadratic minimization problem (the primal form):

$$\min_{w,\xi,\xi^*} \left(\frac{1}{2}\langle w, w\rangle + C(\xi + \xi^*)\cdot\mathbf{1}^{\mathrm{T}}\right), \tag{3}$$

subject to $\xi \geq 0$, $\xi^* \geq 0$,

$$y_i - f(\mathbf{x}_i, \mathbf{w}) \leq \epsilon + \xi_i, \tag{4}$$

$$f(\mathbf{x}_i, \mathbf{w}) - y_i \leq \epsilon + \xi_i^*, \ i = 1, \ldots, n. \tag{5}$$

Here $\xi = (\xi_1, \ldots, \xi_n)$, $\xi^* = (\xi_1^*, \ldots, \xi_n^*)$ are vectors of slack variables representing upper and lower constraints; $\mathbf{1} = (1, \ldots, 1)$ is the unit vector; $C > 0$ is the constant "cost" parameter which specifies the tradeoff between minimization of the risk functional and the smoothness [39]. By using the Lagrangian and the well-known definition of the saddle point conditions, we write the following dual optimization problem [39]:

$$\max_{\alpha, \alpha^*} \Bigg(-\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(\mathbf{x}_i, \mathbf{x}_j) - \epsilon(\alpha + \alpha^*) \cdot \mathbf{1}^{\mathrm{T}} + \mathbf{y} \cdot (\alpha - \alpha^*)^{\mathrm{T}} \Bigg), \tag{6}$$

subject to

$$(\alpha - \alpha^*) \cdot \mathbf{1}^{\mathrm{T}} = 0, \tag{7}$$

$$0 \leq \alpha_i \leq C, \ 0 \leq \alpha_i^* \leq C, \ i = 1, \ldots, n. \tag{8}$$

Here $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ are vectors of optimization variables; $\mathbf{y} = (y_1, \ldots, y_n)$. The optimization problem (6)–(8) is the standard form of the SVR.

2.2. The imprecise SVR

We suppose now that there is an imprecise information about $P(\mathbf{x}, y)$ which is represented as a set of constraints for expectations $\mathbb{E}\psi_j$ of functions ψ_j, $j = 1, \ldots, r$, that is,

$$a_j \leq \int_{\mathcal{X} \times \mathbb{R}} \psi_j(\mathbf{x}, y) \mathrm{d}P(\mathbf{x}, y) \leq b_j, \ j = 1, \ldots, r. \tag{9}$$

So, we have a prior information about data points in the form of r lower $\underline{a}_j = \underline{\mathbb{E}}\psi_j(\mathbf{x})$ and upper $\overline{a}_j = \overline{\mathbb{E}}\psi_j(\mathbf{x})$ expectations of some functions $\psi_j(\mathbf{x})$, $j = 1, \ldots, r$. Here r is a number of indices from the index set $\{1, \ldots, r\}$ of pieces of information. Functions ψ_j depend on the available information about $P(\mathbf{x}, y)$. In particular, the judgment that the ith training example is more important than the k-th example can be represented by means of the function $\psi_j(\mathbf{x}, y) = \mathbf{x}_i - \mathbf{x}_k$ such that the upper expectation of $\psi_j(\mathbf{x}, y)$ is 0, that is, $b_j = \overline{\mathbb{E}}(\mathbf{x}_i - \mathbf{x}_k) = 0$. The probability of the kth point is larger than 0.6 is represented as an interval-valued expectation of the indicator function $\psi_j(\mathbf{x}, y) = I(\mathbf{x} = \mathbf{x}_k)$ taking the value 1 when $\mathbf{x} = \mathbf{x}_k$, that is, $0.6 \leq \mathbb{E}I(\mathbf{x} = \mathbf{x}_k) \leq 1$ or $a_j = 0.6$, $b_j = 1$. The information about the mean value of the kth feature can be represented by means of the function $\psi_j(\mathbf{x}, y) = \mathbf{x}^{(k)}$ and the expectation $\mathbb{E}\mathbf{x}^{(k)}$, where $\mathbf{x}^{(k)}$ is a variable corresponding to the kth feature values.

Denote a set of probability distributions $P(\mathbf{x}, y)$ produced by constraints (9) as $\mathcal{F}$. By returning to the expected risk $R(f)$ in (1), we can say that it belongs to an interval with the lower $\underline{R}$ and upper $\overline{R}$ bounds. In order to use the interval for solving the regression or classification problem, we have to determine a strategy of decision making, which selects one point within this interval for searching optimal parameters of the function f. One of the well-known ways for dealing with the interval-valued expected risk is to use the minimax (pessimistic or robust) strategy for which a distribution $P(\mathbf{x}, y)$ is selected from the set $\mathcal{F}$ such that the expected risk $R(f)$ achieves its largest value or its upper bound $\overline{R}$ for fixed values of parameters of f. In other words, we use the upper bound $\overline{R}$ for computing optimal parameters of f. Since the minimax strategy provides the largest value of the expected risk, then it can be interpreted as an insurance against the worst case because it aims at minimizing the expected loss in the least favorable case [36]. Robust models have been widely exploited in regression and classification problems due to the opportunity to avoid some strong assumptions underlying the standard classification models [53].

The upper bound $\overline{R}$ can be computed by solving the following optimization problem called the natural extension in imprecise probability theory [49]:

$$\overline{R}(f) = \max_{P(\mathbf{x}, y) \in \mathcal{F}} \int_{\mathcal{X} \times \mathbb{R}} l(y, f(\mathbf{x})) \mathrm{d}P(\mathbf{x}, y), \tag{10}$$

subject to (9).

One can see that the problem (10) may be computationally extremely hard because it is assumed that probability distributions from the set $\mathcal{F}$ are arbitrary. Moreover, the set $\mathcal{F}$ may contain the distributions which are unrealistic in a considered applied problem. Therefore, we propose to reduce this set. The reduced set consists of the probability density functions concentrated on elements of the training set S. The density at points which do not belong to the training set is assumed to be 0. Every distribution in this new set can be viewed as a discrete distribution defined on the set S, i.e., we can write $p(\mathbf{x}_i) = p_i$. We denote the set of all discrete distributions $p = (p_1, \ldots, p_n)$ as $\mathcal{P}$. Under some initial conditions for the set $\mathcal{P}$, the distribution $(1/n, \ldots, 1/n)$ may belong to $\mathcal{P}$. In fact, we get a set of empirical expected risk measures such that every risk measure is defined by a distribution p from $\mathcal{P}$. The set of empirical expected risk measures is bounded because the set $\mathcal{P}$ is convex and compact. Therefore, we can deal with the lower and upper bounds for the empirical expected risk.

Proposition 1. *If we have the imprecise information in the form of (9), then the dual optimization problem(6)–(8) for the standard SVR can be rewritten for the imprecise SVR as follows:*

$$\max_{\alpha, \alpha^*, \pi} \Bigg(-\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(\mathbf{x}_i, \mathbf{x}_j) - \epsilon(\alpha + \alpha^*) \cdot \mathbf{1}^{\mathrm{T}} + \mathbf{y} \cdot (\alpha - \alpha^*)^{\mathrm{T}} \Bigg), \tag{11}$$

subject to

$$(\alpha - \alpha^*) \cdot \mathbf{1}^{\mathrm{T}} = 0, \tag{12}$$

$$a_j \leq \langle \pi, \psi_j(\mathbf{x}) \rangle \leq b_j, \ j = 1, \ldots, r, \tag{13}$$

$$\pi \cdot \mathbf{1}^{\mathrm{T}} = 1, \ C \cdot \pi \geq \alpha + \alpha^*. \tag{14}$$

Here $\pi = (\pi_1, \ldots, \pi_n)$, $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ are vectors of optimization variables.

The variables $\pi_1, \ldots, \pi_n$ can be interpreted as the weights of data points. One can see that constraints for vector $\pi = (\pi_1, \ldots, \pi_n)$ coincide with constraints (9). Moreover, the objective function (11) coincides with the objective function (6) of the origin SVR, the constraint (12) coincides with (7). These are very interesting properties of the obtained form of the SVR. We have incorporated constraints producing the set $\mathcal{P}$ into the quadratic programming problem. At that, the dual form of the SVR is added by constraints from the primal form of the linear optimization problem dealing with $\mathcal{P}$. The Lagrangian multipliers α_i, α_i^* are linearly constrained by variables $\pi_1, \ldots, \pi_n$ which correspond to weights of data points. Moreover, it is obvious that if $\alpha_i > 0$, then $\alpha_i^* = 0$, and vice versa. From the above point of view, we can rewrite constraints (14) as follows:

$$C \cdot \pi \geq \alpha, \ C \cdot \pi \geq \alpha^*, \ \forall \pi \in \mathcal{P}. \tag{15}$$

By solving the obtained optimization problem, we find a unique vector of weights assigned to data points, which maximizes the expected risk and takes into account the constraints for weights in the form of $\mathcal{P}$.

It is very interesting that, in fact, we have applied the double duality to the imprecise information about data points. First, we have found the linear dual problem for imprecise data. Second, we have derived the quadratic dual optimization problem with the "dual" imprecise information. As a result, we can observe that the imprecise information is again in its initial form, but it is incorporated into constraints to the quadratic programming problem.

Special Case 1. Suppose that all data points have identical weights, i.e., $a_i = b_i = 1/n,\ \ i = 1, \ldots, n$. Then the optimization problem (11)–(14) is reduced to the programming problem corresponding to the original SVR.

It is important to note that the original SVR is based on the assumption that all examples from a training set have the same weights. The assumption cannot be accepted when the training set is very small. Therefore, it is interesting to consider an extreme case which is viewed as complete ignorance about the weights of examples. In this case, the number r of constraints (9) is 0.

Special Case 2. Suppose that there is complete ignorance about weights of data points. Then the regression model is constructed by using a single example from the training set.

It should be noted that the above special case does not mean that we can replace the training set with a single example. We actually use all examples from the training set, but, in fact, only one of the examples determines the optimal separating function f or its parameters $\mathbf{w}$.

It is well known that an optimal solution to a linear optimization problem can be found among extreme points of the set $\mathcal{P}$ of solutions produced by linear constraints to the problem. Therefore, it is interesting to study whether this property takes place in the considered imprecise SVR.

Special Case 3. The condition $\forall \pi \in \mathcal{P}$ in (15) can be replaced with the condition $\forall \pi \in \mathcal{E}(\mathcal{P})$, where $\mathcal{E}(\mathcal{P})$ is a set of extreme points of $\mathcal{P}$.

We again return to the case of complete ignorance considered above. In this case, the set $\mathcal{P}$ is the unit simplex such that its extreme points have the element 1 at the kth position and 0 at the remaining positions, $k = 1, \ldots, n$. This implies that $\alpha_i + \alpha_i^* = 0$ for all $i \neq j$, and $\alpha_j + \alpha_j^* \leq C$. Since $\alpha_i^* = 0$ if $\alpha_i > 0$, and wise versa, then $\alpha_i = \alpha_i^* = 0$ for all $i \neq j$, and the regression model is constructed by using a single point with the index j.

3. The imprecise SVM for classification

A classification problem statement differs from the regression one by the set of class labels y_i which take only two values -1 and 1, i.e., $y_i \in \{-1, 1\}$. The classification task is to construct an accurate classifier $c : \mathbb{R}^m \to \{-1, 1\}$ that maximizes the probability that $c(\mathbf{x}) = y_i$ for $i = 1, \ldots, n,\ \mathbf{x} \in \mathcal{X}$. One of the ways for solving the problem is to find a real valued separating function $f(\mathbf{x}, w)$ having parameters $\mathbf{w} = (w_0, w_1, \ldots, w_m) \in \mathbb{R}^{m+1}$, for example, $f(\mathbf{x}, \mathbf{w}) = \langle w, \phi(\mathbf{x})\rangle + w_0,\ w = (w_1, \ldots, w_m)$. The sign of the function determines the class label prediction or $c(\mathbf{x})$.

The expected risk in the classification problem can be rewritten as

$$R(f) = \int_{\mathcal{X}\times\{-1,1\}} l(y, f(\mathbf{x}, \mathbf{w}))\mathrm{d}P(\mathbf{x}, y), \tag{16}$$

where the hinge loss function is usually used $l(y, f(\mathbf{x}, \mathbf{w})) = \max(0, 1 - y \cdot f(\mathbf{x}, \mathbf{w}))$. By using again the regularization term $\langle w, w\rangle/2$, the standard SVM classifier can be represented in the form of the following convex optimization problem (the quadratic program) [39]:

$$\min_{\xi, w} R(w) = \min_{\xi, w} \left(\frac{1}{2}\langle w, w\rangle + C\xi \cdot \mathbf{1}^{\mathrm{T}}\right), \tag{17}$$

subject to $\xi \geq 0$

$$y_i(\langle w, \phi(\mathbf{x}_i)\rangle + b) \geq 1 - \xi_i,\ \ i = 1, \ldots, n. \tag{18}$$

Here $\xi = (\xi_1, \ldots, \xi_n)$ is the vector of slack variables. Instead of minimizing the primary objective function (17), a dual objective function, the so-called Lagrangian, can be formed of which the saddle point is the optimum. The dual programming problem is of the form [39]:

$$\max_{\alpha} \left(\alpha \cdot \mathbf{1}^{\mathrm{T}} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)\right), \tag{19}$$

subject to

$$\mathbf{y} \cdot \alpha^{\mathrm{T}} = 0,\ \ 0 \leq \alpha_i \leq C,\ \ i = 1, \ldots, n. \tag{20}$$

After substituting the obtained solution into the expression for the decision function f, we get the "dual" separating function:

$$f(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{21}$$

The parameter b is defined by using support vectors $\mathbf{x}_i$ from the following equation $b = y_j - \sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j)$.

Proposition 2. *Denote $\lambda = (\lambda_1, \ldots, \lambda_n)$. If we have the imprecise information in the form of (9), then the dual optimization problem (19)-(20) for the standard SVM can be rewritten for the imprecise SVM as follows:*

$$\max_{\alpha, \lambda} \left(\alpha \cdot \mathbf{1}^{\mathrm{T}} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), \tag{22}$$

subject to

$$a_j \leq \left\langle \lambda, \psi_j(\mathbf{x})\right\rangle \leq b_j,\ \ j = 1, \ldots, r, \tag{23}$$

$$\lambda \cdot \mathbf{1}^{\mathrm{T}} = 1,\ \ C\lambda \geq \alpha. \tag{24}$$

The objective function does not contain variables λ_i. Moreover, it follows from the constraint $C\lambda_i \geq \alpha_i$ that λ_i can be viewed as the weight of the ith data point, which is determined by $r + 1$ constraints.

Special Case 4. Suppose that all data points have identical weights, i.e., $a_i = b_i = 1/n,\ i = 1, \ldots, n$. Then the optimization problem (22)–(24) is reduced to the programming problem (19)-(20) corresponding to the original SVM.

Special Case 5. Suppose that there is complete ignorance about weights of data points. Then the classification model is constructed by using a single example from the training set.

The proof of Special Cases 4 and 5 are similar to the proof of Special Cases 1 and 2, respectively.

4. One-class classification SVM

4.1. Imprecise extension of Scholkopf's OCC SVM model

According to the OCC classification problem, there are n training examples or observations $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathcal{X}$. The first OCC SVM model proposed by Scholkopf et al. [37,40] aims to construct a function $f(\mathbf{x}, w, \rho)$ with parameters $w = (w_1, \ldots, w_m)$

and ρ, which takes value $+1$ in a "small" region capturing most of the data points from S and -1 elsewhere. It can be done by mapping the data into an alternative higher-dimensional feature space G corresponding to a kernel and by separating them from the origin with maximum margin. Let ϕ be the feature map $\mathbb{R}^m \to G$ which is endowed with an inner product defined as $K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$. So, the OCC SVM aims to find a hyperplane $f(\mathbf{x}, w, \rho) = \langle w, \phi(\mathbf{x}_i) \rangle - \rho = 0$ that separates the data from the origin with maximal margin, i.e., ρ has to be as large as possible so that the volume of the halfspace $\langle w, \phi(\mathbf{x}_i) \rangle \geq \rho$ is minimized. In order to restrict the fraction of input data for which $\langle w, \phi(\mathbf{x}_i) \rangle \leq \rho$, the parameter $\nu \in [0; 1]$ is used. It is analogous to ν used for the ν-SVM [39]. The dual quadratic optimization problem (the Lagrangian) for computing the function f is

$$\min_{\alpha=(\alpha_1,\ldots,\alpha_n)} \left(\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right), \tag{25}$$

subject to

$$0 \leq \alpha_i \leq \frac{1}{\nu n}, \;\; \alpha \cdot \mathbf{1}^{\mathrm{T}} = 1. \tag{26}$$

The function f is defined by n multipliers α_i as

$$f(\mathbf{x}, w, \rho) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho, \tag{27}$$

where the value of ρ can be obtained by using an arbitrary $\mathbf{x}_j$ as $\rho = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}_j)$.

Proposition 3. *Suppose that we have an imprecise information similar to (9) in the form:*

$$a_j \leq \int_{\mathcal{X}} \psi_j(\mathbf{x}) \mathrm{d}P(\mathbf{x}) \leq b_j, \;\; j = 1, \ldots, r. \tag{28}$$

Then the dual quadratic optimization problem(25)-(26) for the standard Scholkopf's OCC SVM can be rewritten for the imprecise OCC SVM as follows:

$$\max_{\alpha,\beta} \left(-\frac{1}{2} \sum_{k=1}^{n} \sum_{t=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right), \tag{29}$$

subject to

$$\alpha \cdot \mathbf{1}^{\mathrm{T}} = 1, \;\; \beta \cdot \mathbf{1}^{\mathrm{T}} = 1, \;\; \alpha \leq \beta/\nu, \tag{30}$$

$$a_j \leq \langle \beta, \psi_j(\mathbf{x}) \rangle \leq b_j, \;\; j = 1, \ldots, r. \tag{31}$$

Here $\beta = (\beta_1, \ldots, \beta_n)$.

We again can see that constraints (31) coincide with constraints (28).

Special Case 6. Suppose that there is complete ignorance about weights of data points. Then the OCC model is constructed by using a single example from the training set.

4.2. Imprecise extension of Tax and Duin's OCC SVM model

Another model of the novelty detection was proposed by Tax and Duin [43]. The main idea underlying the model is to find a sphere with the minimum volume, containing all (or most of) the data points. Since the volume is determined by its radius r or its squared radius r^2, then the volume of the sphere can be minimized by minimizing the radius. On the other hand, the obtained sphere should contain most training objects $\mathbf{x}_i$. A detailed description of the model is given in [43]. We provide the dual optimization problem with Lagrange multipliers $\alpha = (\alpha_1, \ldots, \alpha_n)$

$$\max_{\alpha} \left(\sum_{i=1}^{n} \alpha_i \langle \mathbf{x}_i, \mathbf{x}_i \rangle - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right), \tag{32}$$

subject to

$$0 \leq \alpha_i \leq C, \;\; i = 1, \ldots, n, \;\; \alpha \cdot \mathbf{1}^{\mathrm{T}} = 1. \tag{33}$$

The parameter C gives the trade-off between the volume of the sphere and the number of errors. Incorporating the imprecise information (28) leads to a quadratic optimization problem with the same objective function (32) and the constraints (30)-(31) under condition $C = 1/\nu$.

5. Interval-valued and set-valued data as a special case of the imprecise SVM

It turns out that the interval-valued and set-valued training data can be analyzed in the framework of the imprecise SVM. Let us consider the binary classification problem. Instead of the training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ of point-valued data, we have another training set $S^* = \{(\mathbf{A}_1, y_1), \ldots, (\mathbf{A}_n, y_n)\}$, where $\mathbf{A}_k$ is a matrix representing the set-valued data such that every row of the matrix is the i-th feature vector $\mathbf{x}_k^{(i)}$ belonging to the set-valued example, $i = 1, \ldots, t_k$; t_k is a number of points in the kth set-valued example such that the total number of points is $N = t_1 + \cdots + t_n$ The empirical expected risk in this case is also set-valued with some lower and upper bounds. The upper bound for the expected risk can be written as:

$$\overline{R}(\mathbf{w}) = \max_{\mathbf{x}_k \in \mathbf{A}_k, k=1,\ldots,n} \sum_{i=1}^{n} l(y_i, f(\mathbf{x}_i, \mathbf{w})). \tag{34}$$

Here the expected risk is maximized over all points of set-valued data.

The set-valued uncertainty can be transformed to the probabilistic uncertainty by introducing a set $\mathcal{P}$ of probability distributions π defined on all points of training data. Suppose that t_k points from the kth set-valued observation have indices from an index subset I_k. Then $N = \sum_{k=1}^{n} \sum_{i \in I_k} 1$. If we assume that every set-valued observation has the probability $1/n$ in accordance with the empirical representation of the risk measure, then we can write constraints for π, which determine the set $\mathcal{P}$

$$\sum_{i \in I_k} \pi_i = 1/n, \;\; k = 1, \ldots, n. \tag{35}$$

In other words, we do not know precise probabilities π_i, $i \in I_k$, but we know that the probabilities are restricted by conditions (35). In order to relax too strong conditions (35), we can generalize (35) as follows:

$$a_k \leq \sum_{i \in I_k} \pi_i \leq b_k, \;\; k = 1, \ldots, n, \;\; \sum_{i=1}^{N} \pi_i = 1. \tag{36}$$

The selection of bounds a_k and b_k depends on a considered application and on an imprecise probability model. For example, by using the imprecise Dirichlet model [50] or ε-contaminated model, we can get the following bounds [50]:

$$a_k = \frac{1}{n+s} \leq \sum_{i \in I_k} \pi_i \leq \frac{1+s}{n+s} = b_k, \tag{37}$$

where $s \geq 0$ is the hyperparameter of the Dirichlet model, which determines the influence of the prior distribution on posterior probabilities. In particular, Walley [50] proposes to take $s = 1$ or 2 for many applications.

By taking into account the equality $\varepsilon = s/(n+s)$ for the ε-contaminated model (see details in Section 6.1), we can also write

the bounds a_k and b_k as

$$a_k = \frac{1-\varepsilon}{n} \le \sum_{i \in I_k} \pi_i \le \frac{1+\varepsilon(n-1)}{n} = b_k, \tag{38}$$

where $\varepsilon \in [0, 1]$.

In sum, we write the following minimax optimization problem:

$$\min_{\mathbf{w}} \max_{\pi \in \mathcal{P}} \sum_{k=1}^{n} \sum_{i \in I_k} \pi_i l(y_i, f(\mathbf{x}_i, \mathbf{w})), \tag{39}$$

subject to (36).

The above problem can be represented in the form of the dual optimization problem by using (22)–(24) as follows:

$$\max_{\alpha,\beta} \left(-\frac{1}{2} \sum_{k,t=1}^{n} \sum_{i \in I_k, j \in I_t} \alpha_i \alpha_j y_k y_t K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{k=1}^{n} \sum_{i \in I_k} \alpha_i \right), \tag{40}$$

subject to

$$\sum_{k=1}^{n} \sum_{i \in I_k} \alpha_i y_i = 0, \tag{41}$$

$$\beta \cdot \mathbf{1}^{\mathrm{T}} = 1, \;\; a_k \le \beta_k \le b_k, \;\; k = 1, \ldots, n, \tag{42}$$

$$\beta_k C \ge \sum_{i \in I_k} \alpha_i, \;\; \alpha_i \ge 0, \;\; i \in I_k, \;\; k = 1, \ldots, n. \tag{43}$$

We have derived a way for classifying set-valued training data. The classification procedure by interval-valued data can be implemented in the same way. Every interval of training data is represented as a set of point-valued examples. If numbers of points approximating intervals are $t_1, \ldots, t_n$, then the interval-valued data are classified by means of the problem (40)–(43).

The above derivation of the imprecise SVM by set-valued training data shows that the proposed framework for the imprecisely stated machine learning problems can be regarded as a unified tool for dealing with imprecisions which differ in their types.

6. Numerical examples and some special cases of the incorporated information

6.1. Software reliability growth models as examples of regression models

In order to demonstrate how the additional information incorporated into machine learning algorithms may improve the regression model, we consider an imprecise non-parametric software reliability growth model (SRGM) proposed by Utkin and Coolen [46]. According to [28], software reliability is defined as the probability of failure-free software operation for a specified period of time in a specified environment. One of the ways for analyzing the software reliability is to consider a software testing process, where defects of software are detected and removed. As a result, the software reliability tends to grow. Therefore, in order to estimate the software reliability by using the software testing period, a lot of SRGMs have been developed and successfully verified in many software projects. One of the ways for constructing SRGMs is to use regression models based on the SVR [19]. Although the available SVR-based SRGMs show good learning performance and generalization ability in the software reliability modelling, there are some limitations of their use. These models do not take into account many peculiarities of the software debugging process. Therefore, the main idea underlying the imprecise SRGM is to replace precise weights assigned to the elements of training data by a set of weights produced by means of an intersection of two sets of weights. The first set is produced by a special form of the linear-vacuous mixture or the imprecise ε-contaminated model [49] which can be viewed as a generalization of the well-known ε-contaminated (robust) model [15]. The application of this model relaxes the uniform distribution of weights, but does not assign some precise distribution. The imprecise ε-contaminated model produces the set $\mathcal{P}(\varepsilon)$ of probabilities $\pi = (\pi_1, \ldots, \pi_n)$ such that $\pi_i = (1-\varepsilon)n^{-1} + \varepsilon h_i$, where h_i is arbitrary and $h_1 + \ldots + h_n = 1$, $0 < \varepsilon < 1$ The set $\mathcal{P}(\varepsilon)$ is a subset of the unit simplex $S(1, n)$. It can be produced by $n+1$ hyperplanes

$$\pi_i \ge (1-\varepsilon)n^{-1}, \;\; i = 1, \ldots, n, \;\; \pi_1 + \cdots + \pi_n = 1. \tag{44}$$

The second set is produced by a set of linear comparative equalities

$$\pi_1 \le \pi_2 \le \cdots \le \pi_n, \tag{45}$$

which softly rank the weights. We again do not assign a precise weights for ranking the elements of training data. The second set stems from different importance of data obtained during the debugging process, i.e., it is assumed that elements of the training set at the end of debugging are more important than elements at its beginning because simple errors in a software are removed at the beginning period, a software developer experience during the debugging process may growth and impact on the importance of training data. In order to take into account all factors of the software debugging process, we use the intersection of the sets.

A main difficulty of the model proposed by Utkin and Coolen [46] is that it uses extreme points in order to implement the SVR with the additional imprecise information. The number of extreme points may be very large. Therefore, the algorithm implementing the imprecise SRGM is computationally hard. Moreover, a procedure of the cross-validation in this case becomes also sophisticated. In order to overcome the above difficulties, we apply the imprecise SVR to model the software reliability growth, i.e., we solve the optimization problem (11)-(14). To incorporate the imprecise information into the SVR, we replace constraints (13) in the optimization problem (11)–(14) with the following constraints:

$$\frac{1-\varepsilon}{n} \le \sum_{i=1}^{n} \pi_i I(i = j) \le 1, \;\; j = 1, \ldots, n, \tag{46}$$

$$0 \le \sum_{i=1}^{n} \pi_i (I(i = j) - I(i = j-1)), \;\; j = 2, \ldots, n. \tag{47}$$

Here $I(A)$ is the indicator function taking value 1 if A is true. The first n constraints correspond to the imprecise ε-contaminated model, the second $n-1$ constraints formalize the pairwise comparisons.

We split randomly the data set into three subsets. One of them (training set having n examples) is used to train the model while the other (equal validation and test sets having $n_{val} + n_{test}$ examples) are used to tune the model parameters and to test the model, respectively. For the most real data sets, we use 10% of examples ($n_{val} = 0.1n_0$) for validating and 10% of examples ($n_{test} = 0.1n_0$) for testing. Here $n_0 = n + n_{val} + n_{test}$ is the total number of examples. Every regressor is realized be means of the weighted SVR with parameter of the loss function $\epsilon = 0$.

Since the purpose of these examples is mainly to show the application of the method on simple and easy to visualize problems, the hyperparameters are chosen without fine tuning. For all performed experiments, we quantify the prediction performance with root mean square error measures (MSE) which are defined as

$$MSE = \sqrt{\frac{\sum_{i=1}^{n_{test}} (y_i - f(\mathbf{x}_i))^2}{n_{test}}}. \tag{48}$$

Here y_i, $f(\mathbf{x}_i)$ are the actual and forecasted values, respectively. The corresponding error measures for the standard and the proposed (imprecise) algorithms will be denoted MSE_{st} and MSE_{Imp}.

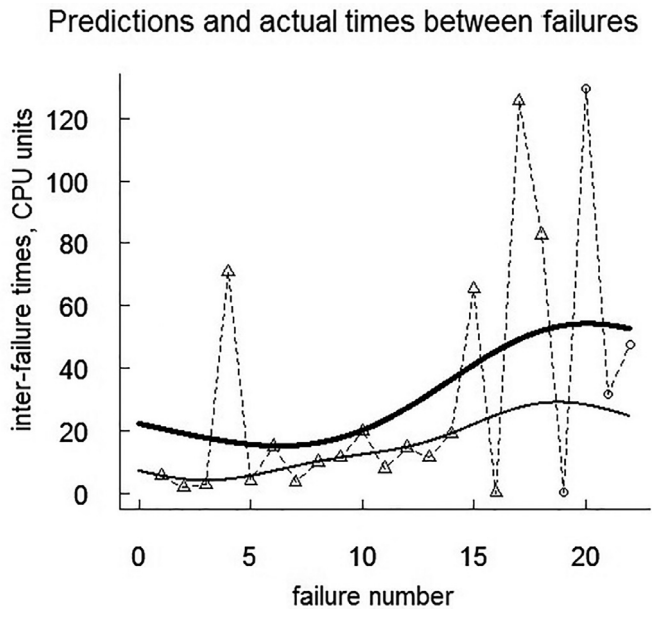

Fig. 1. Fault detection prediction results with two SRGMs and actual results for the first data set (a telemetry network system [32]). The thin and thick curves are the regression functions obtained by means of the standard SVR and the imprecise SVR, respectively.

We will also use the relative absolute difference between MSEs (RAMSE) which is defined as

$$RAMSE = \left|MSE_{\rm st} - MSE_{\rm Imp}\right|/MSE_{\rm st} \times 100. \quad (49)$$

All experiments use a standard Gaussian radial basis function (RBF) kernel

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2/\sigma^2\right) \quad (50)$$

with the kernel parameter σ which determines the geometrical structure of the mapped samples in the kernel space. Different values for the kernel parameter σ and the "cost" parameter C have been tested, choosing those leading to the best results. This procedure is realized by considering all possible values of σ and C in a predefined grid. For every pair of σ and C, the regression function f is constructed by using the proposed imprecise SVR on the basis of the training sets. By using the validation set, the error measure is computed and compared with the error measures obtained by other pairs of σ and C. The smallest error measure corresponds to optimal values of σ and C. The grid for σ is determined as 2^{ν}, where $\nu = -15, \ldots, 0, \ldots, 15$. The values of C are taken in accordance with the expression $C_0 + iC_s$, where $C_0 = 0$ and $C_s = 8$.

The first data set is the software inter-failure times y_i taken from a telemetry network system by AT&T Bell Laboratories published by Pham and Pham [32]. The data set contains 22 observations of the actual time series. An example of the fault detection prediction results with two SRGMs and actual training data (dashed curve) is shown in Fig. 1 where the thin curve corresponds to the non-parametric SRGM based on the standard SVR and the thick curve corresponds to the model using imprecise SVR. The training data are depicted by triangles, the testing data are depicted by circles. One can see from Fig. 1 that the thin curve is close to all training points especially at the beginning of the debugging process. However, it behaves unsatisfactory at the testing period where the large variation of times to failure is observed. At the same time, the thick curve follows up anomalous points. This is carried out due to two reasons. First, the imprecise ε-contaminated model assigns larger weights to anomalous points because they increase the expected risk measure. Second, the comparative information assigns smaller weights to points at the beginning of the debugging process. In spite of the visible improvement of the proposed model, the largest RAMSE is 1.05% by $\varepsilon = 0.1$.

Table 1
A brief introduction about data sets.

	m	n_{-1}	$n - n_{-1}$
PID	8	268	500
Musk	166	499	269
BCWD	30	212	357
Parkinsons	22	147	48
BT	9	36	70
ILP	10	416	167
Seeds	7	70	140
Ionosphere	34	225	126
Ecoli	7	143	193
VC2C	6	210	100
IS	19	90	120

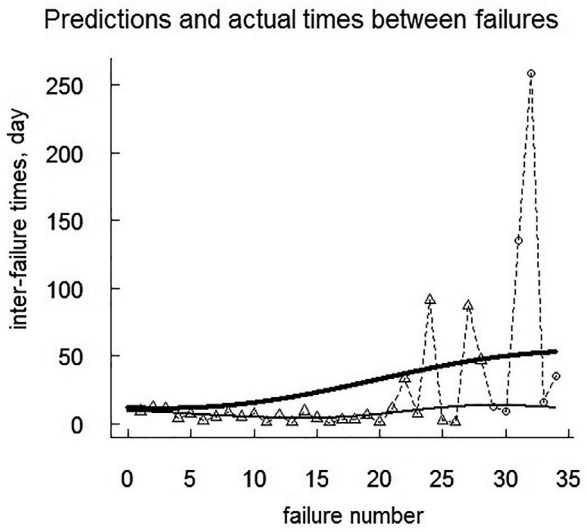

Fig. 2. Fault detection prediction results with two SRGMs and actual results for the second data set (NTDS [18]). The thin and thick curves are the regression functions obtained by means of the standard SVR and the imprecise SVR, respectively.

The second failure data set (NTDS - failure data) was first reported in [18] and contains 34 failure data. An example of the fault detection prediction results with two SRGMs and actual results for the data set is shown in Fig. 2. It can be seen from Fig. 2 that only the last points demonstrate the reliability growth. The largest RAMSE is 15.2% by $\varepsilon = 0.4$.

We have obtained the results similar to those given in [46] for some real software failure data sets. Figs. 1 and 2 almost coincide with the corresponding figures provided in [46] for illustrating the imprecise SRGMs.

6.2. Examples of imprecise classification models

Let us consider data sets which are composed of biomedical measurements from healthy people and people with some disease. The pessimistic strategy is to prefer to attribute a person just in case to people with the disease. At least this person might take additional medical investigation. The "disease" region should be as large as possible. In order to model this requirement, we assume that examples corresponding to people with the disease are more important than other examples. If examples from the "healthy" class are labelled as $y = 1$ and other examples are labelled as $y = -1$, then the weakest condition modelling the

different importance of classes can be written as follows:

$$\sum_{i=1}^{n} \pi_i y_i / n_{y_i} \leq 0. \quad (51)$$

Here n_{y_i} is the number of training elements from the class y_i. Moreover, we again use the imprecise ε-contaminated model and the corresponding constraints (44) with $\varepsilon = 0.1$ in order to relax condition $\pi_i = 1/n,\ i = 1, \ldots, n,$ because this condition may be inconsistent with (51). As a result, we use the intersection of two sets of probabilities. The first one is defined by (51), the second one is produced by the imprecise ε-contaminated model.

The above condition means that the summed mean weight of examples from the "healthy" class is less than the summed mean weight of examples from the class of people with the disease.

We investigate the performance of the proposed model and compare it with the standard SVM by considering two error measures (E and E_{10}), which are the proportion of misclassified examples on a sample of data and the proportion of misclassified examples from the negative class with label $y = -1$. The first measure is usually used to quantify the predictive performance of classification models. However, we are interested in the second measure because it is important for us to minimize errors in one of the classes, namely, in the negative class. The measures can formally be written as

$$E = n_T / n_{test}, \quad E_{10} = n_{01} / n_{test}. \quad (52)$$

Here n_T and n_{01} are numbers of all test examples and negative test examples for which the predicted class for an example does not coincide with its true class, n_{test} is the total number of test data. It should be noted that the measure E_{10} is called very often as the false positive rate.

The proposed algorithm has been evaluated and investigated by the following publicly available data sets: Pima Indian Diabetes (PID), Musk, Breast Cancer Wisconsin Diagnostic (BCWD), Parkinsons, Breast Tissue (BT), Indian Liver Patient (ILP), Seeds, Ionosphere, Ecoli, Vertebral Column 2C (VC2C), Image Segmentation (IS). All data sets are from the UCI Machine Learning Repository [27]. A brief introduction about these data sets are given in Table 1, while a more detailed information can be found from, respectively, the data resources. It should be noted that the first two classes (carcinoma, fibro-adenoma) in the Breast Tissue data set are united and regarded as the negative class. Two classes (disk hernia, spondylolisthesis) in the Vertebral Column 2C data set are also united and regarded as the negative class. The class CYT (cytosolic or cytoskeletal) in Yeast data set is regarded as negative. Other classes are united as the positive class. The first class in Seeds data set is regarded as negative. In the Image Segmentation data set, we unite all classes except for the first class "brickface" into the positive class. The class "brickface" is accepted as negative.

The corresponding classification error measures E and E_{10} for data sets obtained by means of the imprecise SVM and the standard SVM are shown in Table 2. In order to formally show the outperformance of the proposed imprecise model, we apply the t-test which has been proposed and described by Demsar [10] for testing whether the average difference in the performance of two classifiers is significantly different from zero. Since we use the differences between error measures of the standard and imprecise models, then we compare them with 0. The t statistics in this case is distributed according to the Student distribution with $11 - 1$ degrees of freedom. The corresponding p-value for the difference of E_{10} is 0.0353, the 95% confidence interval for the mean of the difference is [0.0045,0.1022]. The t-test demonstrates the clear outperforming of the imprecise model in comparison with the standard one. At the same time, the t-test shows that the difference of error measures E of the models is statistically not significant because the p-value is larger than 0.05 in this case.

Table 2
The classification performance of the imprecise and standard SVMs for real data sets from Table 1.

	Imprecise		Standard	
	E	E_{10}	E	E_{10}
PID	0.368	0.003	0.307	0.238
Musk	0.261	0.159	0.292	0.141
BCWD	0.059	0.019	0.058	0.052
Parkinsons	0.128	0.041	0.131	0.112
BT	0.244	0.123	0.302	0.109
ILP	0.274	0.131	0.289	0.215
Seeds	0.082	0.021	0.077	0.026
Ionosphere	0.295	0.007	0.287	0.068
Ecoli	0.088	0.02	0.090	0.027
VC2C	0.272	0.135	0.261	0.241
IS	0.134	0.041	0.104	0.058

Let us consider the performance of the proposed algorithms with synthetic data having two features x_1 and x_2. Moreover, we use two types of a relational location of data from two classes.

Data sets of Type 1: the training set consisting of two subsets of point-valued data (centers of rectangles) is generated in accordance with the normal probability distributions such that n_{-1} examples (the first class with $y = -1$) are generated with mean values $(m_1,\ m_1)$ and standard deviations $(\sigma_1,\ \sigma_1)$, and n_{+1} examples (the second class with $y = 1$) have mean values $(m_2,\ m_2)$ and standard deviations $(\sigma_2,\ \sigma_2)$. Here we take identical mean values and standard deviations for both features. According to data sets of Type 1, points are concentrated around two centers defined by the corresponding mean values.

Data sets of Type 2: training points are concentrated around two circles having different radius. For generating the points, we use a standard method. First, we generate random radius r_1 for the first class. It is uniformly distributed in the interval $[a_1,\ b_1]$. Then we generate a random angle $\theta \in [0, 2\pi]$. Finally, the random numbers are converted from polar to Cartesian with $(x_1, x_2) = (r_1 \cos\theta, r_1 \sin\theta)$. The same procedure is carried out for the second class, but r_2 is generated in the interval $[a_2,\ b_2]$. The relational location of training examples from different classes is defined by intervals $[a_1,\ b_1]$ and $[a_2,\ b_2]$.

All experiments use the RBF kernel. We perform a cross-validation with 100 repetitions, where in each run, we randomly select $n = 0.75n_0$ training data and $n_{test} = 0.25n_0$ test data, where n_0 is an initial number of all examples. A version of the SVM based on the proposed algorithm has been developed in R.

By generating the data set of Type 2 with parameters $(m_1, m_1) = (3, 3),\ (m_2, m_2) = (6, 6),\ (\sigma_i, \sigma_i) = (2, 2),\ i = 1, 2,$ we study how two separating functions derived by the imprecise SVM and the standard SVM are related to each other depending on the number of training examples. Fig. 3 illustrates three cases of the separating functions by $n = 150$ $(n_{-1} = 75,\ n_{+1} = 75)$, $n = 60$ $(n_{-1} = 30,\ n_{+1} = 30)$, $n = 30$ $(n_{-1} = 15,\ n_{+1} = 15)$. The thick and thin curves correspond to the separating functions obtained by means of the proposed algorithm and the standard SVM, respectively. It can be seen from Fig. 3 that the thick curve "tries" to maximally separate examples from the negative class (small squares) from positive examples (small triangles).

The same study is represented in Fig. 4. In this case, we generate the data sets of Type 2 with parameters $[a_1, b_1] = [0, 0.4]$, $[a_2, b_2] = [0.3, 1]$. We again observe from Fig. 4 that the thick curve "tries" to maximally separate examples from the negative class (small squares) from positive examples (small triangles).

The corresponding classification accuracy measures for data sets of two types obtained by means of the imprecise SVM and the standard SVM are shown in Table 3. It can be seen from Table 3 that the classification error measures E, E_{10} obtained by

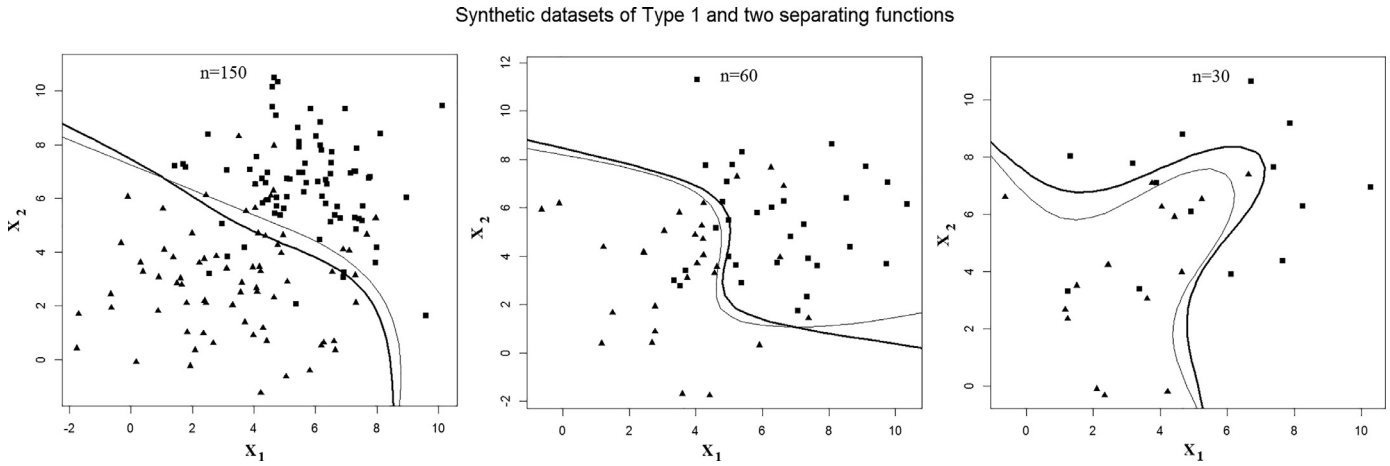

Fig. 3. Separating functions by $n = 150,\ 60,\ 30$ for the data sets of Type 1. The thin and thick curves are the separating functions obtained by means of the standard SVM and the imprecise SVM, respectively.

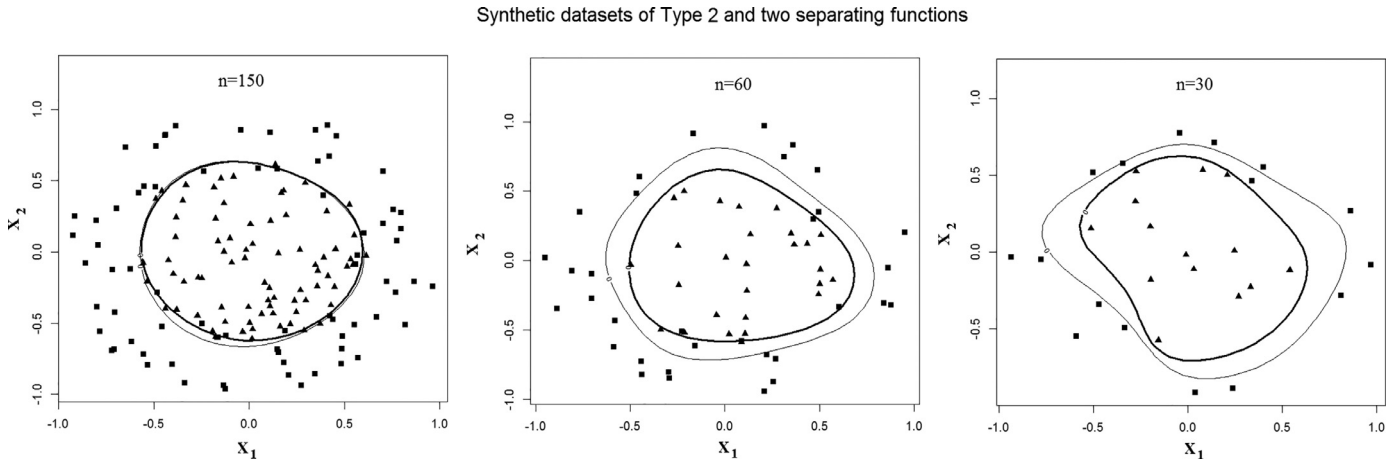

Fig. 4. Separating functions by $n = 150,\ 60,\ 30$ for the data sets of Type 2. The thin and thick curves are the separating functions obtained by means of the standard SVM and the imprecise SVM, respectively.

Table 3
The classification performance of the imprecise and standard SVMs for synthetic data sets of Type 1 and Type 2 by different numbers of training examples ($n = 150, 60, 30$).

n	Data sets of Type 1				Data sets of Type 2			
	Imprecise		Standard		Imprecise		Standard	
	E	E_{10}	E	E_{10}	E	E_{10}	E	E_{10}
150	0.158	0.099	0.159	0.098	0.142	0.066	0.140	0.063
60	0.171	0.088	0.168	0.098	0.148	0.033	0.131	0.040
30	0.184	0.075	0.185	0.105	0.302	0.090	0.285	0.105

using the imprecise SVM and the standard SVM are close to each other when the number of training data is n is rather large (the case $n = 150$). However, one can see that the imprecise SVM outperforms the standard SVM with respect to the error measure E_{10}. It is explained by the constraint (51) which increases the importance of the class $y = -1$.

6.3. Classification models for the structural reliability estimation

Another interesting application of the imprecise classification models where the imprecise information incorporation allows us to robustify the reliability analysis is the structural reliability estimation. The main idea underlying most machine learning approaches in the structural reliability analysis is the improvement of the well known Response Surface Methodology by using the classification procedures for replacing an implicitly defined limit state function with an explicit surrogate function which separates "failure" and "non-failure" regions [2,6,17,35]. The limit state function gives a mathematical definition of the failure event by a certain combination of structure parameters. For positive values of this function, the structure is in a safe or "non-failure" state. Hence, the associated parameter region is referred to as the safe domain. For negative values, the structure is in a failed condition. The associated parameter region is accordingly referred to as the failure domain. The surrogate separating function can be obtained by means of the SVM. One of the important peculiarities of the Response Surface Methodology is that the training set is usually produced by generating samples over the design space using different techniques. As a result, a common assumption of many approaches using the SVM for the structural reliability analysis is that certain distribution functions of random variables characterizing the system reliability behavior and making up the limit state function are known. This use of the SVM is not always justified because its standard form applies the assumption of equal weights for all elements of the training set. Often we have only a set of observations about states of a system. However, the main problem of many SVM-based methods for constructing the limit state function is that the number of the "failure" state observations may be very

small in comparison with the number of "non-failure" states because the "failure" state usually leads to destruction or damage of the analyzed system. One of the ways for dealing with this problem is to use classifiers for imbalanced data [22]. However, a problem here is not the difference between numbers of training examples in two classes, but a very small number of examples in one of the classes. Therefore, for getting a robust limit state function of the system, we propose to consider a set of weights for the "failure" state training examples.

If we denote numbers of training elements corresponding to "failure" and "non-failure" states as n_{-1} and n_1, respectively, then weights of the "non-failure" data are $1/n$, where $n = n_{-1} + n_1$. However, weights of "failure" data are proposed to be imprecise and are defined by the reduced imprecise ε-contaminated model which produces the set $\mathcal{P}(\varepsilon)$ of probabilities $\pi = (\pi_1, \ldots, \pi_{n_{-1}})$ such that $\pi_i = (1-\varepsilon)n^{-1} + \varepsilon h_i$, where h_i is arbitrary under condition $h_1 + \cdots + h_{n_{-1}} = n_{-1}/n$, $0 < \varepsilon < 1$ The set $\mathcal{P}(\varepsilon)$ can be produced by $n+1$ hyperplanes

$$\pi_i \geq (1-\varepsilon)n^{-1}, \ \ i = 1, \ldots, n_{-1}, \ \ \pi_1 + \cdots + \pi_{n_{-1}} = n_{-1}/n. \tag{53}$$

As a result, we get the following forms of constraints (A.86):

$$\frac{1-\varepsilon}{n} \leq \sum_{i=1}^{n_{-1}} \pi_i I(i=j) \leq \frac{1-\varepsilon}{n} + \frac{\varepsilon \cdot n_{-1}}{n}, \ \ j = 1, \ldots, n_{-1}, \tag{54}$$

$$\sum_{i=n_{-1}+1}^{n} \pi_i I(i=j) = \frac{1}{n}, \ \ j = n_{-1}+1, \ldots, n. \tag{55}$$

In order to illustrate the use of the above constraints, we apply an example given by Rajashekhar and Ellingwood [34]. The considered structure is a cantilever beam with a rectangular cross-section and is subjected to a uniformly distributed load. The limit state function is

$$g(x_1, x_2) = 18.461 - 7.477 \times 10^{10} \cdot x_1 / x_2^3. \tag{56}$$

Here x_1 is the load in MPa and x_2 is the depth in mm. x_1 and x_2 are realizations of the normally distributed random variables $X_1 \sim \mathcal{N}(0.001, 0.002)$ and $X_2 \sim \mathcal{N}(250, 37.5)$. We generate 200 observations in accordance with the probability distributions of X_1 and X_2. All observations for which $g(x_1, x_2) < 0$ belong to the failure region. The "failure" and "non-failure" observations are depicted in Fig. 5 by means of small circles and squares, respectively. In order to use the SVM, we multiply the values of the load (x_1) by 250000. RBF kernel with the kernel parameter σ and the "cost" parameter C is used. The optimal parameters are $C = 40000$, $\sigma = 800$. The limit state function (1), the separating functions obtained by means of the standard SVM (2) with $\varepsilon = 0.2$ and the imprecise SVM (3) are shown in Fig. 5. It can be seen from Fig. 5 that the third separating function minimizes the number of misclassified "failure" examples. The error measures of the standard SVM are $E = 0.0027$, $E_{10} = 0.0014$. The error measures of the proposed SVM are $E = 0.0031$, $E_{10} = 0.0005$.

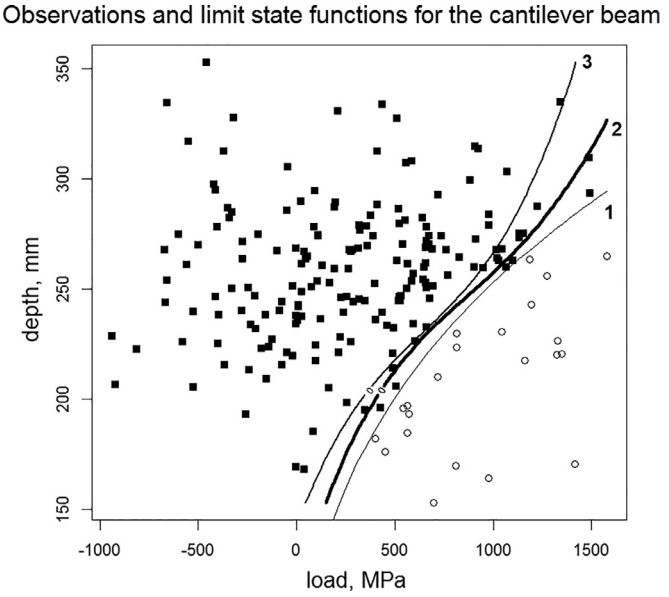

Fig. 5. Exact and approximate limit state functions. Curves with numbers 1,2,3 are the limit state functions obtained from (56), by using the standard SVM and the imprecise SVM, respectively.

6.4. Examples of imprecise OCC models

We study how the imprecise extension of Scholkopf's OCC SVM model can be used with the imprecise information about the mean values of normal observations. A classification error measure used for quantifying the predictive performance of OCC models is defined as

$$E_{\text{OCC}} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (I(y_i \cdot f(\mathbf{x}_i) < 0)). \tag{57}$$

Here y_i is the label of the ith test example $\mathbf{x}_i$; the function f is defined from (27). It should be noted that the labels y_i 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. All experiments use the RBF kernel with the kernel parameter σ.

We consider the performance of the imprecise model with synthetic data having two features x_1 and x_2. A training set consisting of two subsets is generated in accordance with the normal probability distributions such that $n_1 = (1-\gamma_0)n$ examples (the first subset) are generated with mean values $(m_1, m_1) = (3, 3)$, and $n_2 = \gamma_0 n$ examples (the second subset) have mean values $(m_2, m_2) = (8, 8)$. The standard deviation is $\sigma = 2$ for both subsets and both features (see the same generation of data sets of Type 1 in the previous subsection). Here γ_0 is a portion of abnormal examples in the training set. It is 0.1 in most experiments.

An additional imprecise information about mean values of two features is represented as two intervals $[\underline{m}(x_1), \overline{m}(x_1)]$ and $[\underline{m}(x_2), \overline{m}(x_2)]$, respectively. This information can be formally written as

$$\underline{m}(x_k) \leq \sum_{i=1}^{n} \mathbf{x}_i^{(k)} \pi_i \leq \overline{m}(x_k), \ \ k = 1, 2. \tag{58}$$

One can see that the function $\psi_j(\mathbf{x}_i)$ in (31) is $\mathbf{x}_i^{(k)}$, $a_j = \underline{m}(x_k)$ and $b_j = \overline{m}(x_k)$. Moreover, we again use the imprecise ε-contaminated model and the corresponding constraints (44) with $\varepsilon = 0.3$ in order to relax the condition $\pi_i = 1/n$, $i = 1, \ldots, n$.

In order to compare error measures of the proposed model and Scholkopf's OCC SVM model for different numbers of training data n_1 and n_2, we study five cases: $n_1 = 20, 40, 80, 160, 320$. The numbers of generated abnormal data points are $n_2 = 2, 4, 8, 16, 32$, respectively. The parameter of the RBF kernel is 80, the cost parameter ν is 0.5. The incorporated information corresponds to the mean values (m_1, m_1) of generated features for normal data points. In order to get a non-empty set of probability distributions $\mathcal{P}$, we take bounds for mean values $[m_k - 0.1, m_k + 0.1]$, i.e., $a_k = m_k - 0.1$ and $b_k = m_k + 0.1$, $k = 1, 2$. The classification error measures for the proposed model (E_{OCC}) and Scholkopf's OCC SVM model (E_{st}) by different values n_1 and n_2 are shown in Table 4. It can be seen from Table 4 that the additional information in the form of the mean value intervals improves the classification accuracy.

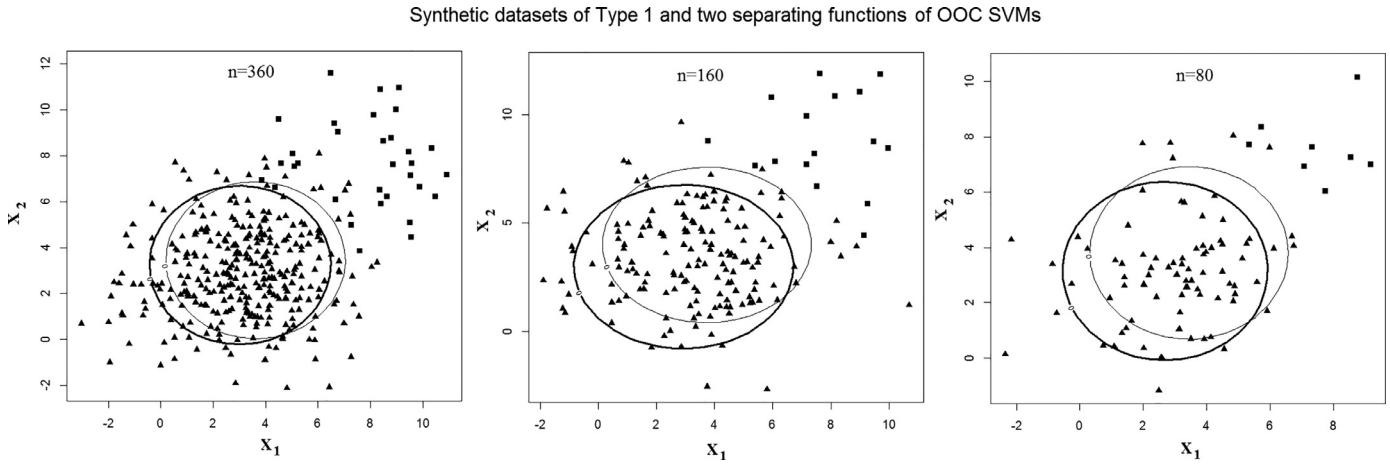

Fig. 6. Separating functions by $n = 80,\ 160,\ 360$ for OCC SVM models. The thin and thick curves are the separating functions obtained by means of the standard and imprecise OCC SVMs, respectively.

Table 4
Error measures obtained by using the imprecise and Scholkopf's OCC SVMs by different values of n_1 for synthetic data sets of Type 1.

n_1	20	40	80	160	320	640
E_{OCC}	0.182	0.209	0.207	0.177	0.192	0.197
E_{st}	0.234	0.225	0.243	0.208	0.225	0.228

Table 5
Error measures obtained by using the imprecise and Scholkopf's OCC SVMs by different values of the parameter ε for synthetic data sets of Type 1.

ε	0.2	0.3	0.4	0.5	0.6	0.7	0.8
E_{OCC}	0.202	0.207	0.212	0.226	0.254	0.286	0.326
E_{st}	0.244	0.244	0.244	0.244	0.244	0.244	0.244

An example of separating functions corresponding to the proposed imprecise model (thick curves) and Scholkopf's OCC SVM model (thin curves) for $n_1 = 80, 160, 360$ is shown in Fig. 6, where generated normal and abnormal points are depicted by small triangles and small squares, respectively. One can see from Fig. 6 that the thick curve is biased toward the mean value point of normal observations. The thick and thin curves correspond to the separating functions obtained by means of the proposed algorithm and the standard SVM, respectively.

Another question is how the parameter ε of the imprecise ε-contaminated model impacts on the classification performance. The corresponding error measures by different values of ε and by $n_1 = 80$, $(m_1, m_1) = (3, 3)$, $(m_2, m_2) = (8, 8)$ are shown in Table 5. It is interesting to see from Table 5 that the proposed model outperforms Scholkopf's OCC SVM model only for $\varepsilon \leq 0.5$. This property is explained by the fact that the additional information in the form of the mean value intervals becomes useless due to a large set of probability distributions produced by the imprecise ε-contaminated model. The value $\varepsilon = 0.1$ is not given in Table 5 because it leads to the empty set $\mathcal{P}$ such that constraints to the quadratic optimization problems become inconsistent. One of the ways to extend the set $\mathcal{P}$ is to increase the mean value interval. In particular, if we take bounds for mean values as $[m_k - 0.4, m_k + 0.4]$, $k = 1, 2$, then the set $\mathcal{P}$ is non-empty by $\varepsilon = 0.1$, and we get $E_{\text{OCC}} = 0.231$.

Finally, we study how the performance of the proposed model depends on distances between mean values $(m_1,\ m_1)$ and $(m_2,\ m_2)$ of normal and abnormal data points, respectively. For

Table 6
Error measures obtained by using the imprecise and Scholkopf's OCC SVMs by different values of m_2 and n_1 for synthetic data sets of Type 1.

n_1	m_2	4	5	6	7	8	9
160	E_{OCC}	0.374	0.334	0.285	0.237	0.188	0.145
160	E_{st}	0.374	0.339	0.298	0.256	0.219	0.184
80	E_{OCC}	0.392	0.352	0.314	0.254	0.207	0.158
80	E_{st}	0.388	0.355	0.316	0.269	0.244	0.207
40	E_{OCC}	0.398	0.363	0.317	0.269	0.209	0.165
40	E_{st}	0.387	0.357	0.315	0.274	0.225	0.196

experiments, we change the mean value m_2 by $n_1 = 40, 80, 160$ and $(m_1, m_1) = (3, 3)$. The corresponding results of experiments are given in Table 6. It can be seen from Table 6 that the incorporated information in the form of the mean value intervals allows us to improve the classification performance for most cases of m_2 and n_1. An exception to the rule is the case of the small n_1 and the very small distance between mean values $(m_1,\ m_1)$ and $(m_2,\ m_2)$, for example, $m_2 - m_1 = 1$ or 2. This is due to impossibility to recognize normal and abnormal examples under these conditions. It can also be seen from Table 6 that the difference $E_{\text{st}} - E_{\text{OCC}}$ increases with m_2, i.e., the incorporated information by large values of m_2 significantly impacts on the OCC SVM performance by means of assigning smaller weights to points which are far from the point $(m_1,\ m_1)$.

Let us consider data sets of Type 2 (see the previous subsection). According to the algorithm used for generating the training examples, normal and abnormal examples have the same mean value (0,0). An information we could apply to data sets of Type 2 is the variance of normal examples which can be computed as a sample variation in accordance with the generating data set. Note that the variance cannot be simply represented as an expectation of some function because it depends on m_k, i.e., $\sigma_k^2 = \mathbb{E}_\pi\left(x^{(k)}\right)^2 - m_k^2$, $k = 1, 2$. However, the knowledge of equal mean values of normal and abnormal examples allows us to simplify the representation of the variance. In this case, we can write

$$\sigma_k^2 = \mathbb{E}_\pi\left(x_1^{(k)}\right)^2 = \sum_{i=1}^{n}\left(\mathbf{x}_i^{(k)}\right)^2\pi_i. \tag{59}$$

To compare error measures of the proposed model and Scholkopf's OCC SVM model for different numbers of training data n_1 and n_2, we study five cases: $n_1 = 20, 40, 80, 160, 320$ (see the same experiments for data sets of Type 1). The numbers of generated abnormal data points are $n_2 = 2, 4, 8, 16, 32$, respectively. The parameter of the RBF kernel is 0.55, the cost parameter v is 0.1.

Table 7
Error measures obtained by using the imprecise and Scholkopf's OCC SVMs by different values of n_1 for synthetic data sets of Type 2.

n_1	20	40	80	160	320	640
E_{OCC}	0.331	0.321	0.294	0.252	0.139	0.110
E_{st}	0.331	0.323	0.311	0.295	0.190	0.152

Table 8
Error measures obtained by using the imprecise and Scholkopf's OCC SVMs by different values of a_2 for synthetic data sets of Type 2.

a_2	0.4	0.5	0.6	0.7	0.8
E_{OCC}	0.298	0.282	0.268	0.229	0.173
E_{st}	0.329	0.307	0.295	0.243	0.179

The parameter of the imprecise ε-contaminated model is 0.3. The classification error measures for the proposed model (E_{OCC}) and Scholkopf's OCC SVM model (E_{st}) by different values n_1 and n_2 are shown in Table 7. It can be seen from Table 7 that the additional information in the form of the variance improves the classification accuracy. Another study with the data sets of Type 2 is how the error measures E_{OCC} and E_{st} depend on the generation parameter a_2 which, in fact, increases the separation of normal and abnormal data points. The corresponding numerical results are given in Table 8. It can be seen from Table 8 that the knowledge of the variance improves the classification quality. However, this improvement is reduced when the normal and abnormal data are totally separated, i.e., the value of a_2 is rather large.

7. Conclusion

A general approach for incorporating imprecise prior knowledge and for robustifying the machine learning SVM-based algorithms has been proposed in the paper. The approach uses the double duality representation in the framework of minimax strategy of decision making, which allows us to get simple extensions of SVMs including additional constraints for optimization variables (the Lagrange multipliers) formalizing the incorporated imprecise information.

It is interesting to note that the objective functions for all optimization problems do not differ from the same functions in the standard SVMs at first view. Moreover, some constraints also coincide in the problems. However, we have to point out that the objective function as well as constraints contain additional variables corresponding to the incorporated information. These variables are implicitly take place in the objective function with zero-valued coefficients, and functions of these variables form the additional constraints.

A drawback of the proposed approach following from the new structure of constraints to the dual optimization problems is that the available standard software for the corresponding SVM-based machine learning algorithms cannot be used. Therefore, a special software has to be developed for implementing the approach. Moreover, the implementation depends on the available incorporated imprecise information.

We have considered only the quadratic form of the SVM optimization problem produced by the L_2-norm regularization term. Other forms of SVMs could be modified in the same way in order to take into account the available prior information. However, this is a direction for further research. We have investigated only one of imprecise models (the imprecise ε-contaminated model) in numerical experiments to provide the robust modification of the standard SVM. However, there are many interesting imprecise models, for example, the imprecise pari-mutuel model [49] , the constant odds-ratio (π, ε) model [49], the Kolmogorov–Smirnov confidence limits for the empirical cumulative distribution function [21], which could be successfully applied to development of the robust SVM modifications. However, this is also a direction for further research.

Acknowledgement

We would like to express our appreciation to the anonymous referees whose valuable comments have improved the paper.

The reported study was partially supported by RFBR, research project No. 17-01-00118.

Appendix

Proof of Proposition 1. It should be noted that the optimization problem (10) is linear, and the following dual optimization problem can be written:

$$\overline{R}(f) = \min_{c_0, c_i, d_i} \left\{ c_0 + \sum_{j=1}^{r} (c_i b_i - d_i a_i) \right\}, \tag{60}$$

subject to $c_0 \in \mathbb{R}$, c_i, $d_i \geq 0$, $i = 1, \ldots, n$,

$$c_0 + \sum_{j=1}^{r} \left(c_j - d_j\right) \psi_j(\mathbf{x}, y) \geq l(y, f(\mathbf{x}, \mathbf{w})), \ \mathbf{x} \in \mathcal{X}. \tag{61}$$

Here c_0, c_i, d_i are optimization variables such that c_0 corresponds to the constraint $\int_{\mathcal{X}} \mathrm{d}P(\mathbf{x}, y) = 1$, c_i corresponds to the constraint $\int_{\mathcal{X}} \psi_i(\mathbf{x}, y)\mathrm{d}P(\mathbf{x}, y) \leq b_i$, and d_i corresponds to the constraint $a_i \leq \int_{\mathcal{X}} \psi_i(\mathbf{x}, y)\mathrm{d}P(\mathbf{x}, y)$. A detailed discussion of the dual problem and its meaning can be found in Walley's book [49].

The restriction of $\mathcal{F}$ leads to a change only of constraints (61) in the problem (60)-(61). They can be rewritten now as follows:

$$c_0 + \sum_{j=1}^{r} \left(c_j - d_j\right) \psi_j(\mathbf{x}_i, y_i) \geq l(y_i, f(\mathbf{x}_i, \mathbf{w})), \quad i = 1, \ldots, n. \tag{62}$$

The dual form has a very important property: the objective function has to be minimized. We get rid of the maximization problem. Denote

$$A_i(c, d) = c_0 + \sum_{j=1}^{r} \left(c_j - d_j\right) \psi_j(\mathbf{x}_i, y_i), \tag{63}$$

$$B(c, d) = c_0 + \sum_{j=1}^{r} \left(c_j \overline{a}_j - d_j \underline{a}_j\right). \tag{64}$$

Here $c = (c_0, c_1, \ldots, c_r)$, $d = (d_1, \ldots, d_r)$.

After substituting the ϵ-insensitive loss function into (62), the imprecise minimax regression problem statement can be finally rewritten as the following minimization problem:

$$\min_{w} \overline{R}(f) = \min_{w, c, d} B(c, d), \tag{65}$$

subject to c_i, $d_i \geq 0$, $i = 1, \ldots, n$, and

$$A_i(c, d) \geq y_i - f(\mathbf{x}_i, \mathbf{w}) - \epsilon, \tag{66}$$

$$A_i(c, d) \geq -y_i + f(\mathbf{x}_i, \mathbf{w}) - \epsilon, \tag{67}$$

$$A_i(c, d) \geq 0, \ i = 1, \ldots, n. \tag{68}$$

By adding the standard Tikhonov regularization term $\frac{1}{2}\langle w, w\rangle$ to the objective function (65), we rewrite the objective function (65) as follows:

$$\overline{R}(\mathbf{w}) = \frac{1}{2}\langle w, w\rangle + C \cdot B(c, d). \tag{69}$$

We get the quadratic programming problem. Instead of minimizing the primary objective function (69), a dual objective function, the so-called Lagrangian, can be formed of which the saddle point is the optimum. The Lagrangian is

$$\begin{aligned} L = {} & \frac{1}{2}\langle w, w\rangle + C \cdot B(c, d) - \sum_{j=1}^{r} c_j \eta_j - \sum_{j=1}^{r} d_j \mu_j \\ & - \sum_{i=1}^{n} \beta_i A_i(c, d) - \sum_{i=1}^{n} \alpha_i (A_i(c, d) - y_i + f(\mathbf{x}_i, \mathbf{w}) + \epsilon) \\ & - \sum_{i=1}^{n} \alpha_i^* (A_i(c, d) + y_i - f(\mathbf{x}_i, \mathbf{w}) + \epsilon). \end{aligned} \tag{70}$$

Here $\eta_j,\ \mu_j,\ j = 1, \ldots, r,\ \alpha_i, \alpha_i^*, \beta_i,\ i = 1, \ldots, n,$ are Lagrange multipliers. Hence, the dual variables have to satisfy positivity constraints $\eta_j \geq 0,\ \mu_j \geq 0,\ \alpha_i \geq 0,\ \alpha_i^* \geq 0,\ \beta_i \geq 0$ for all i and j. The saddle point can be found by setting the derivatives equal to zero

$$\partial L / \partial w_0 = -(\alpha - \alpha^*) \cdot \mathbf{1}^{\mathrm{T}} = 0, \tag{71}$$

$$\partial L / \partial w_j = w_j - \left\langle \mathbf{x}^{(j)}, \alpha + \alpha^* \right\rangle = 0,\ \ j = 1, \ldots, m, \tag{72}$$

$$\partial L / \partial c_0 = C - (\alpha + \alpha^* + \beta) \cdot \mathbf{1}^{\mathrm{T}} = 0, \tag{73}$$

$$\partial L / \partial c_j = C \cdot b_j - \eta_j - \left\langle \alpha + \alpha^* + \beta, \psi_j(\mathbf{x}) \right\rangle = 0,\ \ j = 1, \ldots, r, \tag{74}$$

$$\partial L / \partial d_j = -C \cdot a_j - \mu_j + \left\langle \alpha + \alpha^* + \beta, \psi_j(\mathbf{x}) \right\rangle = 0,\ \ j = 1, \ldots, r. \tag{75}$$

Here $\mathbf{x}^{(j)}$ is the jth element of the vector $\mathbf{x}$. Denote $\pi = (\alpha + \alpha^* + \beta)/C$. Then the Lagrangian can be simplified and is written as (11)-(14). □

Proof of Special Case 1. The function $\psi_i(\mathbf{x})$ takes the value 1 if $\mathbf{x} = \mathbf{x}_i$ and 0 if $\mathbf{x} \neq \mathbf{x}_i$. Then we get the following constraints:

$$\pi_i = 1/n,\ \ \alpha_i \leq C/n,\ \ \alpha_i^* \leq C/n,\ \ i = 1, \ldots, n. \tag{A.76}$$

It can be seen from the above that the Lagrangian multipliers α_i, α_i^* do not depend on π_i. □

Proof of Special Case 2. In this case, constraints (13) are absent. We have only the trivial constraint for π given in (14). This implies that π may be an arbitrary point in the n-dimensional unit simplex. If we return to the primal optimization problem (69), then it can be simplified for the case of complete ignorance as follows:

$$\overline{R}(\mathbf{w}) = \frac{1}{2}\langle w, w\rangle + C \cdot c_0, \tag{A.77}$$

subject to

$$c_0 \geq \max\left(0, |y_i - f(\mathbf{x}_i, \mathbf{w})| - \epsilon\right),\ \ i = 1, \ldots, n. \tag{A.78}$$

It is obvious that the optimal value of c_0 is

$$c_0 = \max\left(0, \max_{i=1,\ldots,m} |y_i - f(\mathbf{x}_i, \mathbf{w})| - \epsilon\right). \tag{A.79}$$

If there are points outside the region defined by the parameter ϵ, i.e., $|y_i - f(\mathbf{x}_i, \mathbf{w})| - \epsilon > 0$, then the variable c_0 is determined by the largest difference $|y_i - f(\mathbf{x}_i, \mathbf{w})| - \epsilon$. In other words, the regression model is constructed by using a single example from the training set. □

Proof of Special Case 3. Every vector $\pi = (\pi_1, \ldots, \pi_n)$ of $\mathcal{P}$ can be expressed through the extreme points as

$$\pi_i = \sum_{k=1}^{t} \lambda_k \cdot q_i^{(k)},\ \ i = 1, \ldots, n. \tag{A.80}$$

Here $q_i^{(k)}$ is the ith element of the kth extreme point; λ_k is a weight of the kth extreme point, $\sum_{k=1}^{t} \lambda_k = 1$, t is the total number of extreme points. Then

$$\alpha + \alpha^* \leq C \cdot \pi = C \sum_{k=1}^{t} \lambda_k \cdot q_i^{(k)}. \tag{A.81}$$

The condition $\forall \pi \in \mathcal{P}$ means that the constraint $\alpha + \alpha^* \leq C \cdot \pi$ can be rewritten as $\alpha + \alpha^* \leq C \cdot \min_{\pi \in \mathcal{P}} \pi$. Suppose that there exists some smallest value of π_i from $\pi \in \mathcal{P}$. However, this value cannot be larger than $\min_{k=1,\ldots,t} q_i^{(k)}$ due to the above expression. This implies that

$$\alpha_i + \alpha_i^* \leq C \cdot \min_{k=1,\ldots,t} q_i^{(k)}, \tag{A.82}$$

as was to be proved. □

Proof of Proposition 2. It is important to note that expressions (60)–(64) introduced for the regression problem are not changed for the classification task. However, constraints (68) depend on the used loss function. As a result, we rewrite these constraints for every $i = 1, \ldots, n$, as follows:

$$A_i(c, d) \geq 1 - y_i f(\mathbf{x}_i, \mathbf{w}),\ \ A_i(c, d) \geq 0. \tag{A.83}$$

The upper expected risk added the standard Tikhonov regularization term is of the form (69) in this case. However, the Lagrangian is

$$\begin{aligned} L = {} & \frac{1}{2}\langle w, w\rangle + C \cdot B(c, d) \\ & - \sum_{i=1}^{n} \alpha_i (A_i(c, d) - 1 + y_i \langle w, \phi(\mathbf{x}_i)\rangle + y_i w_0) \\ & - \sum_{j=1}^{r} c_j \eta_j - \sum_{j=1}^{r} d_j \mu_j - \sum_{i=1}^{n} \beta_i A_i(c, d). \end{aligned} \tag{A.84}$$

Here $\eta_j,\ \mu_j,\ j = 1, \ldots, r,\ \alpha_i,\ \beta_i,\ i = 1, \ldots, n,$ are Lagrange multipliers. Hence, the dual variables have to satisfy positivity constraints $\eta_j \geq 0,\ \mu_j \geq 0,\ \alpha_i \geq 0,\ \beta_i \geq 0$ for all i and j.

Conditions for the Lagrange multipliers are similar to the corresponding conditions (71)–(75) in the proof of Proposition 1. After substituting the obtained conditions into the Lagrangian, we finally get the dual optimization problem with the objective function (22) and constraints

$$\langle y, \alpha\rangle = 0, \tag{A.85}$$

$$C \cdot a_j \leq \sum_{i=1}^{n} \left\langle \alpha + \beta, \psi_j(\mathbf{x}) \right\rangle \leq C \cdot b_j,\ \ j = 1, \ldots, r, \tag{A.86}$$

$$(\alpha + \beta) \cdot \mathbf{1}^{\mathrm{T}} = C. \tag{A.87}$$

Let us denote $\lambda_i = (\alpha_i + \beta_i)/C$. Then we rewrite the last constraints (A.86)-(A.87) as (23)-(24). □

Proof of Proposition 3. Note that (60) is not changed for the OCC SVM, but (61) can be rewritten as

$$\min_{c_0, c, d, w, \rho} \left\{ c_0 + \sum_{k=1}^{n} b_k c_k - \sum_{k=1}^{n} a_k d_k \right\} - \rho, \tag{A.88}$$

subject to

$$c_0 + \sum_{j=1}^{r} \left(c_j - d_j\right)\psi_j(\mathbf{x}) \geq l(f(\mathbf{x}), w, \rho), \ \ \mathbf{x} \in \mathcal{X}. \tag{A.89}$$

The upper bound for the expected risk with the Tikhonov regularization term is computed as follows:

$$\overline{R}(w) = \min\left(\frac{1}{2}\langle w, w\rangle + \frac{1}{\nu}\left\{c_0 + \sum_{k=1}^{n} b_k c_k - \sum_{k=1}^{n} a_k d_k\right\} - \rho\right), \tag{A.90}$$

subject to $c_k \geq 0$, $d_k \geq 0$, $k = 1, \ldots, n$, and (62).

Let us write the Lagrangian by using the hinge-loss function

$$\begin{aligned} L = &\frac{1}{2}\langle w, w\rangle + \frac{1}{\nu}B(c, d) - \rho \\ &- \sum_{k=1}^{n}\lambda_k c_k - \sum_{k=1}^{n}\mu_k d_k - \sum_{k=1}^{n}\beta_k A_k(c, d) \\ &- \sum_{k=1}^{n}\alpha_k(A_k(c, d) - \rho + \langle w, \phi(\mathbf{x}_i)\rangle). \end{aligned} \tag{A.91}$$

Here $\lambda_k \geq 0$, $\mu_k \geq 0$, $\beta_k \geq 0$, $\alpha_k \geq 0$, $k = 1, \ldots, n$, are Lagrange multipliers. After simplifying with using the saddle point, we can write the final dual quadratic optimization problem (29)–(43). □

Proof of Special Case 6. The primal optimization problem (A.88)-(A.89) can be rewritten as

$$\min_{c_0, w, \rho}(c_0 - \rho), \tag{A.92}$$

subject to

$$c_0 = \max\left\{0, \rho - \min_{i=1,\ldots,n}\langle w, \phi(\mathbf{x}_i)\rangle\right\}. \tag{A.93}$$

Hence we get the unconstrained problem

$$\max_{w,\rho}\min\left\{\rho, \min_{i=1,\ldots,n}\langle w, \phi(\mathbf{x}_i)\rangle\right\}. \tag{A.94}$$

We again make decision by using a single point. □

References

[1] A. Antonucci, R. de Rosa, A. Giusti, F. Cuzzolin, Temporal data classification by imprecise dynamical models, in: Proceedings of the 8th International Symposium on Imprecise Probability: Theories and Applications, SIPTA, Compiegne, France, 2013, pp. 13–22.

[2] A. Basudhar, S. Missoum, A.H. Sanchez, Limit state function identification using support vector machines for discontinuous responses and disjoint failure domains, Probab. Eng. Mech. 23 (1) (2008) 1–11.

[3] A. Ben-Tal, S. Bhadra, C. Bhattacharyya, J.S. Nath, Chance constrained uncertain classification via robust optimization, Math. Program. Ser. B 127 (1) (2011) 145–173.

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

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

[6] J.M. Bourineta, F. Deheegera, M. Lemaire, Assessing small failure probabilities by combined subset simulation and support vector machines, Struct. Saf. 33 (6) (2011) 343–353.

[7] C. Bouveyron, S. Girard, Robust supervised classification with mixture models: Learning from data with uncertain labels, Pattern Recognit. 42 (11) (2009) 2649–2658.

[8] M. Chavent, F. de A. T., Y. Lechevallier, R. Verde, New clustering methods for interval data, Comput. stat. 21 (2) (2006) 211–229.

[9] R.M.C.R. de Souza, F. de A. T., Clustering of interval data based on city-block distances, Pattern Recognit. Lett. 25 (2004) 353–365.

[10] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[11] T.N. Do, F. Poulet, Kernel methods and visualization for interval data mining, in: Internaional Symposium on Applied Stochastic Models and Data Analysis, ASMDA, 5, 2005, pp. 345–355.

[12] G.M. Fung, O.L. Mangasarian, J.W. Shavlik, Knowledge-based support vector machine classifiers, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, USA, 2002, pp. 521–528.

[13] Y. Gao, F. Gao, Edited adaboost by weighted knn, Neurocomputing 73 (16–18) (2010) 3079–3088.

[14] L.E. Ghaoui, G.R.G. Lanckriet, G. Natsoulis, Robust classification with interval data, Technical report report no. ucb/csd-03-1279, University of California, Berkeley, California, 2003. 94720.

[15] P.J. Huber, Robust Statistics, Wiley, New York, 1981.

[16] E. Hullermeier, Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization, Int. J. Approx. Reason. 55 (7) (2014) 1519–1534.

[17] J.E. Hurtado, D.A. Alvarez, Classification approach for reliability analysis with stochastic finite-element modeling, J. Struct. Eng. 129 (8) (2003) 1141–1149.

[18] Z. Jelinski, P.B. Moranda, Software reliability research, in: W. Greiberger (Ed.), Statistical Computer Performance Evaluation, Academic Press, New York, 1972, pp. 464–484.

[19] C. Jin, S.W. Jin, Software reliability prediction model based on support vector regression with improved estimation of distribution algorithms, Appl. Soft Comput. 15 (2014) 113–120.

[20] T. Joachims, Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[21] N.L. Johnson, F. Leone, Statistics and Experimental Design in Engineering and the Physical Sciences, 1, Wiley, New York, 1964.

[22] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, Handling imbalanced datasets: a review, GESTS Int. Trans. Comput. Sci. Eng. 30 (1) (2006) 25–36.

[23] G.R.G. Lanckriet, L.E. Ghaoui, C. Bhattacharyya, M.I. Jordan, A robust minimax approach to classification, J. Mach. Learn. Res. 3 (2002) 555–582.

[24] F. Lauer, G. Bloch, Incorporating prior knowledge in support vector machines for classification: a review, Neurocomputing 71 (7–9) (2008) 1578–1594.

[25] G.Y. Li, V. Jeyakumar, G.M. Lee, Robust conjugate duality for convex optimization under uncertainty with application to data classification, Nonlinear Anal. Theory Methods Appl. 74 (6) (2011) 2327–2341.

[26] Y. Li, D. de Ridder, R.P.W. Duin, M.J.T. Reinders, Integration of prior knowledge of measurement noise in kernel density classification, Pattern Recognit. 41 (2008) 320–330.

[27] M. Lichman, UCI machine learning repository, 2013. http://archive.ics.uci.edu/ml/.

[28] M.R. Lyu, Handbook of Software Reliability Engineering, McGraw-Hill, New York, 1996.

[29] O.L. Mangasarian, Knowledge-based linear programming, SIAM J. Optim. 15 (2) (2005) 375–382.

[30] E.A.L. Neto, F.A.T. de Carvalho, Centre and range method to fitting a linear regression model on symbolic interval data, Comput. Stat. Data Anal. 52 (2008) 1500–1515.

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

[32] L. Pham, H. Pham, Software reliability models with time-dependent hazard function based on Bayesian approach, IEEE Trans. Syst. Man Cybern. Part A 30 (1) (2000) 25–35.

[33] F. Provost, T. Fawcett, Robust classification for imprecise environments, Mach. Learn. 42 (3) (2001) 203–231.

[34] M.R. Rajashekhar, B.R. Ellingwood, A new look at the response surface approach for reliability analysis, Struct. Saf. 12 (3) (1993) 205–220.

[35] B. Richard, C. Cremona, L. Adelaide, A response surface method based on support vector machines trained with an adaptive experimental design, Struct. Saf. 39 (11) (2012) 14–21.

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

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

[38] B. Scholkopf, P. Simard, A. Smola, V. Vapnik, Prior knowledge in support vector kernels, in: Proceedings of the 1997 conference on Advances in neural information processing systems, 10, MIT Press, Cambridge, 1998, pp. 640–646.

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

[40] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, Support vector method for novelty detection, in: Advances in Neural Information Processing Systems, 2000, pp. 526–532.

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

[42] Z. Sun, Z.K. Zhang, H.G. Wang, Incorporating prior knowledge into kernel based regression, Acta Automatica Sinica 34 (12) (2008) 1515–1521.

[43] D. Tax, R. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.

[44] A.N. Tikhonov, V.Y. Arsenin, Solution of ill-posed problems, 1977. W. H. Winston, VH Winston & Sons; Washington DC.

[45] T.B. Trafalis, R.C. Gilbert, Robust support vector machines for classification and computational issues, Optim. Methods Softw. 22 (1) (2007) 187–198.

[46] L.V. Utkin, F.P.A. Coolen, A robust weighted SVR-based software reliability growth model, Reliab. Eng. Syst. Saf. 176 (2018) 93–101.
[47] L.V. Utkin, Y.A. Zhuk, Imprecise prior knowledge incorporating into one-class classification, Knowl. Inf. Syst. 41 (1) (2014) 53–76.
[48] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[49] P. Walley, Statistical Reasoning with Imprecise Probabilities, Chapman and Hall, London, 1991.
[50] P. Walley, Inferences from multinomial data: Learning about a bag of marbles, J. R. Stat. Soc. Ser. B 58 (1996) 3–57. With discussion.
[51] W. Wang, Z. Xu, W. Lu, X. Zhang, Determination of the spread parameter in the Gaussian kernel for classification and regression, Neurocomputing 55 (3) (2003) 643–663.
[52] X. Wang, P.M. Pardalos, A survey of support vector machines with uncertainties, Ann. Data Sci. 1 (3–4) (2014) 293–309.
[53] H. Xu, C. Caramanis, S. Mannor, Robustness and regularization of support vector machines, J. Mach. Learn. Res. 10 (7) (2009) 1485–1510.
[54] X. Yang, Q. Song, Y. Wang, A weighted support vector machine for data classification, Int. J. Pattern Recognit. Artif. Intell. 21 (5) (2007) 961–976.
[55] Z. Zhao, P. Zhong, Y. Zhao, Learning svm with weighted maximum margin criterion for classification of imbalanced data, Math. Comput. Model. 54 (3–4) (2011) 1093–1099.

Lev V. Utkin is currently a Professor at the Telematics Department (under the Central Scientific Research Institute of Robotics and Technical Cybernetics) in Peter the Great St.Petersburg Polytechnic University. In 1986 he graduated from St.Petersburg State Electrotechnical University (former Leningrad Electrotechnical Institute). He holds a Ph.D. in Information Processing and Control Systems (1989) from the same university and a D.Sc. in Mathematical Modelling (2001) from St.Petersburg State Institute of Technology, Russia. He worked as the Vice-rector for Research and the Head of the Department of Control, Automation and System Analysis from 2006 till 2015 in St.Petersburg State Forest Technical University. Author of more than 300 scientific publications. His research interests are focused on machine learning, imprecise probability theory, reliability analysis, decision making, bioinformatics.