Imprecise imputation as a tool for solving classification problems with mean values of unobserved features

Lev V. Utkin*, Yulia A. Zhuk
Department of Control, Automation and System Analysis
St.Petersburg State Forest Technical University
Institutski per. 5, 194021, St.Petersburg, Russia
e-mail: lev.utkin@gmail.com

Abstract

A method for solving a classification problem when there is only partial information about some features is proposed. This partial information comprises the mean values of features for every class and the bounds of the features. In order to maximally exploit the available information a set of probability distributions is constructed such that two distributions are selected from the set which define the minimax and minimin strategies. Random values of features are generated in accordance with the selected distributions by using Monte Carlo technique. As a result, the classification problem is reduced to the standard model which is solved by means of the support vector machine. Numerical examples illustrate the proposed method.

Keywords: data mining; classification; minimax strategy; imputation; Monte Carlo; support vector machine; hinge loss function.

1 Introduction

There are several major data mining techniques including classification, clustering, novelty detection, etc. We consider classification as a data mining technique used to predict an unobserved output value y based on an observed input vector $\mathbf{x}$. This requires us to estimate a predictor f from training data or a set of example pairs of $(\mathbf{x}, y)$. A special very important problem of the statistical machine learning is the binary classification problem which can be regarded as a task of classifying some objects into two classes (groups) in accordance with their properties or features. In other words, we have to classify each pattern $\mathbf{x}$ into one of the classes by means of a discriminant function f.

*Corresponding author, e-mail: lev.utkin@gmail.com

A common assumption in supervised learning is that training and predicted data are drawn from the same (unknown) probability distribution, i.e., training and predicted data come from the same statistical model. As a result, most machine learning algorithms and methods exploit this assumption which, unfortunately, does not often hold in practice. This may lead to a performance deterioration in the induced classifiers [1, 2]. This problem may arise if we have imbalanced data [18] or in case of rare events or observations [36]. The assumption does not hold also in case of partially known or observed features. For instance, it may take place when we know only some mean values of the features, but can not get their actual values during training.

One of the approaches to handle the above problem and to cope with the imbalance and possible inconsistencies of training and predicted data is the minimax strategy for which the classification parameters are determined by minimizing the maximum possible risk of misclassification [1, 2]. This is an "extreme" strategy of decision making. As pointed out in [1], the minimax classifiers may be seen as over-conservative since its goal is to optimize the performance under the least favorable conditions. Therefore, it is interesting to simultaneously study the so-called minimin or optimistic strategy for which the classification parameters are determined by minimizing the minimum possible risk of misclassification. This is another "extreme" strategy.

By taking into account the above, we propose a classification model using the minimax and minimin strategies for situations when a part of features are observed and there are precise values of the features corresponding to different classified classes, but our initial information about other part of features is restricted by mean values of the features for every class. In other words, we know only mean values (expectations) of some features and do not have any observations. This is a very restrictive information which should be exploited. The features with this information will be called unobserved for simplicity. A typical example of the above situation is a mode of production of reinforced concrete beams whose quality and strength depend on a number of parameters such as the weight of reinforcement bars, concrete materials, etc. If we have not observed or measured some of the parameters before, it is difficult to reject new beams or to classify them into two classes: defective (rejected) or of high quality, because we do not have the learning set of beams with the measured parameters. However, if we know, for instance, how much steel has been used up by manufacturing N beams, then we are able to evaluate the average weight of steel in a beam. The information can be elicited, for instance, from experts. Often, it is easy for experts to provide judgments about some average values of a feature for every class because this information is the most simple and understandable.

One of the simplest ways to solve the classification problem with the partial information is to assume that the mean values are observed values. In fact, we replace in this case an unknown probability distribution of data of a feature by the deterministic variable which takes one value corresponding to the mean value of this feature. Of course, we accept here a very strong assumption which may lead to a significant performance deterioration especially if the underlying

probability distribution is not symmetric. Another way is to find the mean values of every observed feature and use a simplest classification algorithm considered by many authors, for instance, by [26]. However, we lose some useful information in this case, which can be inferred from the observations.

In order to maximally exploit the available information about features, we propose another approach whose underlying ideas can be formulated a combination of multiple imputation [24] and imprecise models of features.

As indicated in [25], imputation is a class of methods by which an estimation of the missing value or its distribution is used to generate predictions from a given model. In particular, either a missing value is replaced with an estimation of the value or alternatively the distribution of possible missing values is estimated and corresponding model predictions are combined probabilistically. Various imputation treatments for missing values in training data are available that may be deployed at prediction time [3, 9, 14, 15, 20, 21]. However, some treatments such as multiple imputation [24] are particularly suitable to induction. In particular, multiple imputation (or repeated imputation) is a Monte Carlo approach that generates multiple simulated versions of a data set such that each are analyzed and the results are combined to generate inference.

We do not know the probability distributions of data for unobserved features. However, the mean values of features and their boundary values produce a set of probability distributions bounded by some lower and upper cumulative distribution functions (CDFs). This way leads to constructing the so-called p-boxes [6, 12] from data. It should be noted that the considered set of distributions is not the set of parametric distributions having the same parametric form as the bounding distributions, but it is the set of all possible distributions restricted by the lower and upper bounds. This is an important feature of the proposed approach in this paper. A probability distribution is selected from the p-box in order to make a pessimistic decision, which maximizes the risk function as a measure of the classification error. In other words, the well-known minimax strategy is applied for solving the classification problem, which appears as an insurance against the worst case [23]. Another probability distribution is selected from the p-box in order to make an optimistic or minimin decision. The similar idea applied to regression models has been considered in [30, 31, 32]. So, the first idea is to consider the lower and upper probability distributions of feature data produced by the corresponding mean values and bounds for the feature values.

It should be noted that the obtained bounding probability distributions do not belong to standard types of probability distributions and their convolution for combining features and for computing parameters of the discriminant function is an extremely hard problem. Therefore, in order to cope with this problem the second idea is proposed. We can apply the Monte Carlo technique for generating random values of features, which are governed by the probability distributions selected from the p-boxes in accordance with the minimax and minimin strategies [4, 27]. In a nutshell, for every example of the training set, we generate a (large) number of random values for unobserved features. It is a multiple imputation technique which has been applied to classification prob-

lems [10, 25, 37]. But the main distinction of the proposed approach from the available ones is that it is based on some partial information about unobserved features and uses the p-boxes for generating random values of features.

After carrying out this procedure, the classification problem can be solved by means of standard methods, for instance, by means of the support vector machine (SVM). The Monte Carlo technique has also been applied to general classification problems [7, 28]. It has been successfully applied to reliability analysis problems in the framework of classification models [16, 17]. Of course, the Monte Carlo technique requires additional computation efforts. However, its main advantage is its simplicity. Moreover, we get the standard classification problem solved by standard available software tools.

We have to stress that there is no sense in applying the proposed model when we have some missing values among the observed values of a feature. The model has to be used when we do not have observations for some features at all and only mean values of the features and their bounds are known.

The paper is organized as follows. A statement of the well-known standard classification problem is given in Section 2. This statement is extended on the case of a set of probability distributions of training data in Section 3. In this section, two strategies: minimax and minimin, are formally introduced. The classification problem with mean values for a part of unobserved features is considered in Section 4. A general method for constructing the classification model by partial information about some features with using the set of probability distributions is described in the same section. A question of the training data generation for realizing Monte Carlo simulation is solved in Section 5. A way for reducing the classification problem with partial information about features to the standard problem and its solution by means of the SVM method is given in Section 6. Numerical examples with synthetic data and with the real datasets, including Iris, Pima Indian Diabetes, Mammographic masses, Parkinsons, Indian Liver Patient, Breast Cancer Wisconsin (Original), Breast Cancer Wisconsin (Diagnostic), Musk, Lung-cancer datasets from UCI Machine Learning Repository [13], are provided in Section 7.

2 The standard classification problem

The binary-classification problem can be formulated as follows. There are predictor-response data with a binary response y representing the observation of classes $y = -1$ and $y = 1$. The binary-classification problem is to estimate a region in predictor space in which class 1 is observed with the greatest possible majority. Suppose we are given empirical data

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n) \in \mathbb{R}^l \times \{-1, +1\}. \tag{1}$$

Here $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ is some nonempty set of the patterns or examples; $y_1, ..., y_n$ are labels or outputs taking the values -1 and $+1$; l is the number of features. It is supposed that the number of elements in the training set

belonging to the class y is n_y and their indices form the set of indices $N(y)$, i.e., we can write $n_{-1} + n_1 = n$ and $N(y) = \{i : y_i = y\}$.

Classification problem is usually characterized by an unknown CDF $F_0(\mathbf{x}, y)$ on $\mathbb{R}^l \times \{-1, +1\}$ defined by the training set or examples $\mathbf{x}_i$ and their corresponding class labels y_i.

The main problem is to find a decision function $g(\mathbf{x})$, which predicts accurately the class label y of any example $\mathbf{x}$ that may or may not belong to the training set. In other words, we seek a function g that minimizes the classification error, which is given by the probability that $g(\mathbf{x}) \neq y$. One of the possible approaches for solving the problem is the discriminant function approach which uses a real valued function $f(\mathbf{x})$ called the discriminant function whose sign determines the class label prediction: $g(\mathbf{x}) = \mathrm{sgn}(f(\mathbf{x}))$. The discriminant function $f(\mathbf{x})$ may be parametrized with some parameters $\mathbf{w} = (w_0, w)$, $w = (w_1, ..., w_l)$, that are determined from the training examples by means of a learning algorithm. In particular, the function $f(\mathbf{x})$ may be linear, i.e., $f(\mathbf{x}) = \langle w, \mathbf{x} \rangle + w_0$. Introduce also the notation $x_i^{(k)}$ for the i-th element of the vector $\mathbf{x}_k$.

Given the training data the linear discriminant training problem is to minimize the following risk measure [33]:

$$R(\mathbf{w}) = \int_{\mathbb{R}^l \times \{-1,1\}} L(\mathbf{x}, y) \mathrm{d}F_0(\mathbf{x}, y).$$

Here the loss function $L(\mathbf{x}, y)$ usually takes a non-zero value when the sign of the discriminant function (the class label prediction) does not coincide with the class label y. The minimization of the risk measure is carried out over the parametric class of functions $f(\mathbf{x})$. In other words, the function $f(\mathbf{x})$ provides the minimum of $R(\mathbf{w})$ such that $R(\mathbf{w}_{\mathrm{opt}}) = \min_{\mathbf{w}} R(\mathbf{w})$.

3 The classification problem under a set of probability distributions

Let us represent the joint probability as $F_0(\mathbf{x}, y) = F_0(\mathbf{x} \mid y) \cdot P(y)$. Here $P(y)$ is the prior probability that an example $\mathbf{x}$ belongs to the class y. Then we can rewrite the risk measure taking into account two values of y

$$R(\mathbf{w}) = P(-1)R_{-1}(\mathbf{w}) + P(1)R_1(\mathbf{w}).$$

Here

$$R_{-1}(\mathbf{w}) = \int_{\mathbb{R}^l} L(\mathbf{x}, -1) \mathrm{d}F_0(\mathbf{x} \mid -1), \tag{2}$$

$$R_1(\mathbf{w}) = \int_{\mathbb{R}^l} L(\mathbf{x}, 1) \mathrm{d}F_0(\mathbf{x} \mid 1). \tag{3}$$

By assuming that features are independent, we can rewrite the above risk measures as

$$R_y(\mathbf{w}) = \int_{\mathbb{R}^l} L(\mathbf{x}, y) \prod_{i=1}^{l} \mathrm{d}F_i(x_i \mid y), \ y = -1, 1.$$

Suppose that the distributions F_i are unknown. However, we assume that some lower and upper bounds for a set $\mathcal{F}_i(y)$ of the CDFs $F_i(x \mid y)$ are known accurate to w, and they are $\underline{F}_i(x \mid y)$ and $\overline{F}_i(x \mid y)$, respectively. We can write

$$\mathcal{F}_i(y) = \{F_i(x \mid y) \mid \underline{F}_i(x \mid y) \leq F_i(x \mid y) \leq \overline{F}_i(x \mid y)\}.$$

In other words, there is an unknown precise "true" CDF $F_i(x \mid y) \in \mathcal{F}_i(y)$ for every $y \in \{-1, +1\}$ and every $i = 1, ..., l$, but we do not know it and only know that it belongs to the set $\mathcal{F}_i(y)$. It has been mentioned that the set $\mathcal{F}_i(y)$ is not the set of parametric distributions having the same parametric form as the bounding distributions, but it is the set of all possible distributions restricted by the lower and upper bounds.

3.1 The minimax strategy

One of the possible strategies to derive an estimator is the minimax (pessimistic) strategy. According to the minimax strategy, we select a CDF from the set $\mathcal{F}_i(-1)$ and a CDF from the set $\mathcal{F}_i(1)$ such that the risk measures $R_{-1}(\mathbf{w})$ and $R_1(\mathbf{w})$ achieve their maximum for every fixed $\mathbf{w}$. The minimax strategy can be explained in a simple way. We do not know a precise CDF F_i and every CDF from $\mathcal{F}_i(y)$ can be selected. Therefore, we should take the "worst" distribution providing the largest value of the risk measure. The minimax criterion appears as an insurance against the worst case because it aims at minimizing the expected loss in the least favorable case [23].

Denote $\mathcal{F}(y) = \mathcal{F}_1(y) \times ... \times \mathcal{F}_l(y)$. Since the sets $\mathcal{F}(-1)$ and $\mathcal{F}(+1)$ are obtained independently for $y = -1$ and $y = 1$, respectively, then

$$\overline{R}(\mathbf{w}) = \max_{F(\mathbf{x} \mid y) \in \mathcal{F}(y)} R(\mathbf{w}) = \sum_{y=-1,1} P(y) \max_{F(\mathbf{x} \mid y) \in \mathcal{F}(y)} R_y(\mathbf{w}).$$

The minimax risk functional with respect to the minimax strategy is now of the form:

$$\overline{R}(\mathbf{w}_{\text{opt}}) = \min_{\mathbf{w}} \overline{R}(\mathbf{w}).$$

Let us consider in detail the first problem $\max_{F(\mathbf{x} \mid -1) \in \mathcal{F}(-1)} R_{-1}(\mathbf{w})$. Most loss functions $L(\mathbf{x}, -1)$ applied in classification are increasing with f. This implies that the upper bound for $R_{-1}(\mathbf{w})$, i.e., the maximum of $R_{-1}(\mathbf{w})$ over all distributions from $\mathcal{F}(-1)$ is achieved at the CDFs $\underline{F}(\mathbf{x} \mid -1)$ (see, for instance, Walley's paper [35]). Hence, there holds

$$\overline{R}_{-1}(\mathbf{w}) = \int_{\mathbb{R}^l} L(\mathbf{x}, -1) \mathrm{d}\underline{F}(\mathbf{x} \mid -1).$$

Here

$$\underline{F}(\mathbf{x} \mid y) = \prod_{i=1}^{l} \widetilde{F}_i(x_i \mid y),$$

where

$$\widetilde{F}_i(x_i \mid y) = \begin{cases} \underline{F}_i(x_i \mid y), & L(\mathbf{x}, y) \text{ increases with } x_i, \\ \overline{F}_i(x_i \mid y), & L(\mathbf{x}, y) \text{ decreases with } x_i. \end{cases} \tag{4}$$

The above condition can be rewritten in terms of the function f instead of L

$$\widetilde{F}_i(x_i \mid -1) = \begin{cases} \underline{F}_i(x_i \mid -1), & f(\mathbf{x}) \text{ increases with } x_i, \\ \overline{F}_i(x_i \mid -1), & f(\mathbf{x}) \text{ decreases with } x_i, \end{cases}$$

$$\widetilde{F}_i(x_i \mid 1) = \begin{cases} \underline{F}_i(x_i \mid 1), & f(\mathbf{x}) \text{ decreases with } x_i, \\ \overline{F}_i(x_i \mid 1), & f(\mathbf{x}) \text{ increases with } x_i. \end{cases}$$

In the same way, we can consider the second problem $\max_{F(\mathbf{x} \mid 1) \in \mathcal{F}(1)} R_1(\mathbf{w})$. Most loss functions $L(\mathbf{x}, 1)$ are decreasing with f. Therefore, the upper bound for $R_1(\mathbf{w})$ is achieved at the distribution $\overline{F}(\mathbf{x} \mid 1)$. This implies that

$$\overline{R}_1(\mathbf{w}) = \int_{\mathbb{R}^l} L(\mathbf{x}, 1) \mathrm{d}\overline{F}(\mathbf{x} \mid 1).$$

Finally, we get the upper bound for the risk measure $R(\mathbf{w})$, which is of the form:

$$\overline{R}(\mathbf{w}) = \sum_{y=-1,1} P(y) \int_{\mathbb{R}^l} L(\mathbf{x}, y) \prod_{i=1}^{l} \mathrm{d}\widetilde{F}_i(x_i \mid y).$$

Now we have two tasks. First, we have to define CDFs $\underline{F}_i(x \mid y)$ and $\overline{F}_i(x \mid y)$ from the available information for every $y = -1, 1$ and for every $i = 1, ..., l$. Second, we have to define the prior probabilities of classes $P(-1)$ and $P(1)$.

3.2 The minimin strategy

The minimin strategy can be regarded as a direct opposite of the minimax strategy. According to the minimin strategy, the risk measure R is minimized over all probability distributions from the set $\mathcal{F}$ as well as over all values of parameters. The strategy can be called optimistic because it selects the "best" probability distribution from the set $\mathcal{F}$. Of course, the minimin strategy is of little interest. Nevertheless, we study it in order to compare "extreme" cases (minimax and minimin strategies).

Similarly to the minimax strategy, we can write

$$\underline{R}(\mathbf{w}) = \min_{F(\mathbf{x} \mid y) \in \mathcal{F}(y)} R(\mathbf{w}) = \sum_{y=-1,1} P(y) \min_{F(\mathbf{x} \mid y) \in \mathcal{F}(y)} R_y(\mathbf{w}).$$

Since loss functions $L(\mathbf{x}, -1)$ applied in classification are increasing with f, then the lower bound for $R_{-1}(\mathbf{w})$, i.e., the minimum of $R_{-1}(\mathbf{w})$ over all distributions from $\mathcal{F}(-1)$ is achieved at the distribution $\overline{F}(\mathbf{x} \mid -1)$. The loss function $L(\mathbf{x}, 1)$ is decreasing. Therefore, the lower bound for $R_1(\mathbf{w})$ is achieved at the distribution $\underline{F}(\mathbf{x} \mid 1)$. Hence, there holds

$$\underline{R}(\mathbf{w}) = \sum_{y=-1,1} P(y) \int_{\mathbb{R}^l} L(\mathbf{x}, y) \prod_{i=1}^{l} \mathrm{d}\widehat{F}_i(x_i \mid y).$$

where

$$\widehat{F}_i(x_i \mid y) = \begin{cases} \underline{F}_i(x_i \mid y), & L(\mathbf{x}, y) \text{ decreases with } x_i, \\ \overline{F}_i(x_i \mid y), & L(\mathbf{x}, y) \text{ increases with } x_i. \end{cases} \tag{5}$$

The optimization problem for computing the optimal values of parameters $\mathbf{w}$ for the minimin strategy can be written as

$$\underline{R}(\mathbf{w}_{\text{opt}}) = \min_{\mathbf{w}} \underline{R}(\mathbf{w}).$$

4 Mean values of features and a method for constructing the model

Suppose that an object is characterized by l features. Moreover, we have the training set (1). Every observation $\mathbf{x}_i$ contains the observed values of t features $x_1^{(i)}, ..., x_t^{(i)}$. We assume that features with numbers $1, ..., t$ are observed without loss of generality. However, other $l - t$ features are unobserved, and we know only conditional mean values $m_i(y)$ of the features for every class and their bounds a_i and b_i, $i = t + 1, ..., l$. How to classify the objects in this case?

One of the simplest ways is to assume that the mean values are observed values. In other words, we can write $x_j^{(i)} = m_j(y)$, $j = t + 1, ..., l$, for all $i = 1, ..., n$, i.e., for all observations. This way can be applied when there are a lot of observation. However, when the amount of statistical data is small, the above replacement of observations by mean values may lead to incorrect classification. Moreover, we do not take into account the information about bounds of feature values here, which might be useful.

Another way is to find the mean values of every observed feature with the number $j = 1, ..., t$, for every class as

$$m_j(-1) = \frac{1}{n_{-1}} \sum_{i:y_i=-1} x_j^{(i)}, \; m_j(1) = \frac{1}{n_1} \sum_{i:y_i=1} x_j^{(i)}.$$

Then we can exploit the simplest classification algorithm considered by many authors, for instance, by [26]. The algorithm is based on analyzing the distances between a predicted vector $\mathbf{x}$ and two vectors of mean values of features. The smallest distance determines the class of $\mathbf{x}$. It has been noted in [26] that the proposed decision is the best we can do if we have no prior information about the probabilities of the two classes. However, we lose some useful information in this case, which can be inferred from the observations.

Therefore, we have to develop a classification method which maximally exploits the available information about features.

The first important assumption we use below is that the values of t observed features are governed by the nonparametric or empirical distribution.

By dealing with the unobserved features, we consider two cases or two important assumptions. The first one is that we have conditional expectations $m_j(y)$ defined for every class. The second one is that we have unconditional expectation for every feature, which does not depend on the class. This case

is less informative, but it is typical for many applications. It is reduced to the first case by accepting the equality $m_j(-1) = m_j(1)$.

Let us divide the discriminant function into two parts:

$$f^{(1)}(\mathbf{x}^{(1)}) = w_0 + \sum_{i=1}^{t} w_i x_i = w_0 + \left\langle \mathbf{x}^{(1)}, w^{(1)} \right\rangle,$$

$$f^{(2)}(\mathbf{x}^{(2)}) = \sum_{i=t+1}^{l} w_i x_i = \left\langle \mathbf{x}^{(2)}, w^{(2)} \right\rangle.$$

Here $\mathbf{x}^{(1)} = (x_1, ..., x_t)$, $\mathbf{x}^{(2)} = (x_{t+1}, ..., x_l)$, $w^{(1)} = (w_1, ..., w_t)$, $w^{(2)} = (w_{t+1}, ..., w_l)$.

The whole discriminant function is the sum $f^{(1)}(\mathbf{x}^{(1)}) + f^{(2)}(\mathbf{x}^{(2)})$. We assume that every function $f^{(1)}$ and $f^{(2)}$ has some conditional CDFs $F_1(f^{(1)} \mid y)$ and $F_2(f^{(2)} \mid y)$ for every $y = -1, 1$, respectively.

Let us return to the risk measure $R_y(\mathbf{w})$ defined in (2) and (3). It can be rewritten as follows:

$$R_y(\mathbf{w}) = \int_{\mathbb{R}^2} L(f^{(1)} + f^{(2)}, y) \mathrm{d}F^{(1)}(f^{(1)} \mid y) \mathrm{d}F^{(2)}(f^{(2)} \mid y)$$

$$= \int_{\mathbb{R}^{l-t+1}} L(f^{(1)} + f^{(2)}, y) \mathrm{d}F^{(1)}(f^{(1)} \mid y) \prod_{j=t+1}^{l} \mathrm{d}\widetilde{F}_j(x_j \mid y).$$

Here $F_j(x_j \mid y)$ is the conditional CDF of the j-th feature for the class y.

By assuming that the observed features are governed by the empirical distribution, we can conclude that the distribution of the function $f^{(1)}$ is also empirical, i.e., its PDF is the weighted sum of Dirac functions $\delta\left(f^{(1)} - f_i^{(1)}\right)$ with weights $1/n_y$. Hence, we obtain

$$R_y(\mathbf{w}) = \int_{\mathbb{R}^{l-t+1}} L(f, y) \frac{1}{n_y} \sum_{i \in N(y)} \delta\left(x - f_i^{(1)}\right) \mathrm{d}x \prod_{j=t+1}^{l} \mathrm{d}\widetilde{F}_j(x_j \mid y)$$

$$= \frac{1}{n_y} \sum_{i \in N(y)} \int_{\mathbb{R}^{l-t}} L(f_i^{(1)} + f^{(2)}, y) \prod_{j=t+1}^{l} \mathrm{d}\widetilde{F}_j(x_j \mid y).$$

The precise CDFs F_j, $j = t+1, ..., l$, are unknown. However, we know the mean values of every feature with numbers $t+1, ..., l$ for every class y and the bounds of their values. Therefore, we can construct a set of CDFs with some lower and upper bounds. Given the mean value $m_i(y)$ of the i-th feature and its bounds a_i, b_i, the lower $\underline{F}_i(x \mid y)$ and upper $\overline{F}_i(x \mid y)$ conditional CDFs of the i-th feature values are

$$\underline{F}_i(x \mid y) = \begin{cases} 0, & x < a_i, \\ \max\left(0, \frac{x - m_i(y)}{x - a_i}\right), & a_i \le x < b_i, \\ 1, & x \ge b_i, \end{cases}$$

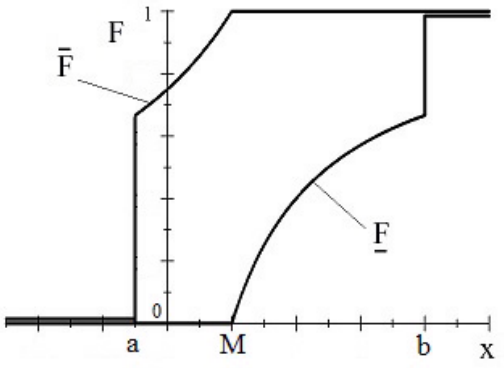

Figure 1: The lower and upper probability distributions

$$\overline{F}_i(x \mid y) = \begin{cases} 0, & x < a_i, \\ \min\left(1, \frac{b_i - m_i(y)}{b_i - x}\right), & a_i \le x < b_i, \\ 1, & x \ge b_i. \end{cases}$$

It should be noted that the expression for the upper bound $\overline{F}_i(x \mid y)$ can be obtained by using the natural extension [19, 34] which can be represented as the following linear programming problem:

$$\overline{F}_i(x \mid y) = \min_{c,d} \left(c + d \cdot m_i(y)\right),$$

subject to $c, d \in \mathbb{R}$, $c + d \cdot z \ge \mathbf{1}\{z \le x\}$, $\forall z \in [a_i, b_i]$.

Here $\mathbf{1}\{z \le x\}$ is the indicator function taking the value 1 if $z \le x$. The lower bound $\underline{F}_i(x \mid y)$ can be obtained in the same way by solving the following programming problem:

$$\underline{F}_i(x \mid y) = \max_{c,d} \left(c + d \cdot m_i(y)\right),$$

subject to $c, d \in \mathbb{R}$, $c + d \cdot z \le \mathbf{1}\{z \le x\}$, $\forall z \in [a_i, b_i]$.

The same bounds have been differently obtained in the work [11].

The lower and upper CDFs are shown in Fig. 1, where $M = m(y) = 2$, $a = -1$, $b = 8$. The resulting bounds are optimal in the sense that they could not be any tighter under the given information. However, this does not mean that any distribution whose CDF is inscribed within this bounded probability region would have the same expectations $m_i(y)$. The obtained set is more rich and produces the p-box. This leads to a more conservative and cautious solution of the classification problem.

Now we have two problems. The first one is to determine the CDFs F_j, $j = t + 1, ..., l$. The second problem is to solve an optimization problem for computing parameters $\mathbf{w}$ by using the above expressions for the risk measure.

Since the function $L(f,-1)$ is increasing, then the upper bound for $R_{-1}(\mathbf{w})$ can be written as

$$\overline{R}_{-1}(\mathbf{w}) = \frac{1}{n_{-1}} \sum_{i \in N(-1)} \int_{\mathbb{R}^{l-t}} L(f_i^{(1)} + f^{(2)}, -1) \prod_{j=t+1}^{l} \mathrm{d}\widetilde{F}_j(x_j \mid -1). \quad (6)$$

Here the upper bound $\overline{R}_{-1}(\mathbf{w})$ depends only on the bounds for CDFs $\widetilde{F}_j(x_j \mid -1)$. This is a very important property which will be used later.

The function $L(f,1)$ is decreasing. This implies that the upper bound for $R_1(\mathbf{w})$ is

$$\overline{R}_1(\mathbf{w}) = \frac{1}{n_1} \sum_{i \in N(1)} \int_{\mathbb{R}^{l-t}} L(f_i^{(1)} + f^{(2)}, 1) \prod_{j=t+1}^{l} \mathrm{d}\widetilde{F}_j(x_j \mid 1). \quad (7)$$

Here the upper bound $\overline{R}_1(\mathbf{w})$ depends also only on the bounds for CDFs $\widetilde{F}_j(x_j \mid 1)$.

It should be noted that it is difficult to integrate in (6)-(7) in an explicit form in order to get some functions of parameters w even for the simplest loss functions L. However, we can apply the standard Monte-Carlo technique. By using this technique, random values of features with the indices $j = t+1, ..., l$, are generated in accordance with the CDFs $\widetilde{F}_j(x_j \mid 1)$ for the class $y = 1$ and with the CDFs $\widetilde{F}_j(x_j \mid -1)$ for the class $y = -1$. By generating K_i random vectors of features $\mathbf{x}_{i,k}^{(2)} = (x_{t+1}^{(i,k)}, ..., x_l^{(i,k)})$, $k = 1, ..., K_i$, for every $i = 1, ..., N(y)$ and every $y = -1, 1$ in accordance with the CDF $\widetilde{F}_j(x_j \mid -1)$ and the CDF $\widetilde{F}_j(x_j \mid 1)$, we rewrite (6)-(7) as follows:

$$\overline{R}_{-1}(\mathbf{w}) = \frac{1}{n_{-1}} \sum_{i \in N(-1)} \frac{1}{K_i} \sum_{k=1}^{K_i} L(f_i^{(1)} + \left\langle \mathbf{x}_{i,k}^{(2)}, w^{(2)} \right\rangle, -1),$$

$$\overline{R}_1(\mathbf{w}) = \frac{1}{n_1} \sum_{i \in N(1)} \frac{1}{K_i} \sum_{k=1}^{K_i} L(f_i^{(1)} + \left\langle \mathbf{x}_{i,k}^{(2)}, w^{(2)} \right\rangle, 1).$$

Finally, we obtain the upper risk measure as a function of parameters w as

$$\overline{R}(\mathbf{w}) = \sum_{y=-1,1} \frac{P(y)}{n_y} \sum_{i \in N(y)} \frac{1}{K_i} \sum_{k=1}^{K_i} L(f_i^{(1)} + \left\langle \mathbf{x}_{i,k}^{(2)}, w^{(2)} \right\rangle, y_i), \quad (8)$$

where $x_j^{(i,k)} \sim \widetilde{F}_j(x_j \mid -1)$ for $i \in N(-1)$ and $x_j^{(i,k)} \sim \widetilde{F}_j(x_j \mid 1)$ for $i \in N(1)$.

In fact, we extend the training set by generating the "missing" values of features. We reduce the learning problem with combined types of the training information to the standard problem when there are training data in the form of real and generated observations of all features. It is important to note that we do not replace here the "missing" features by their mean values $m_i(-1)$, $i = t+1, ..., l$. The "missing" values are replaced by a set of random values of features generated in accordance with the corresponding lower and upper CDFs.

The optimization problem for computing parameters w for the minimin strategy is of the same form as (8). However, the value $x_j^{(i,k)}$ is governed by the CDF $\widehat{F}_j(x_j \mid -1)$ for $i \in N(-1)$ and $x_j^{(i,k)}$ is governed by the CDF $\widehat{F}_j(x_j \mid 1)$ for $i \in N(1)$. This is just one distinguish of optimization problems by the minimax and minimin strategies.

An important question is how to determine the functions $\widetilde{F}_j$ and $\widehat{F}_j$ or how to determine the type of dependence between L and x_j. We can propose two possible ways for doing that. First, the dependence can be determined by experts or by a decision maker on the basis of a preliminary analysis of features and classes. Very often, we can evaluate how possible changes of the feature values impact on the output variable y on the basis of physical meaning of the analyzed classification problem. Of course, this way is simple, but, generally, it can not be always applied to classification problems. Second, we can enumerate 2^{l-t+1} variants of the CDFs $\widetilde{F}_j$ and $\widehat{F}_j$ by taking different lower and upper CDFs instead of $\widetilde{F}_j$ and $\widehat{F}_j$. In accordance with the minimax strategy, the optimal risk measure is the largest value of the risk measure $R(\mathbf{w})$ by optimal parameters $\mathbf{w}_{\text{opt}}$. The same procedure can be applied to the minimin strategy. However, we search for the smallest value of the risk measure $R(\mathbf{w})$ by optimal parameters $\mathbf{w}_{\text{opt}}$ in this case.

5 A procedure for generation of random feature values

Let us consider how to generate random feature values in accordance with the above CDFs. First, we analyze the lower CDF. It can be seen from its form that the corresponding random variable is concentrated on two subsets. The first subset is the interval from $m_i(y)$ till b_i. The second is the point b_i. The probability that the random variable is in the interval $[m_i(y), b_i)$ is equal to $(b_i - m_i(y))/(b_i - a_i)$. The probability of the point b_i is $(m_i(y) - a_i)/(b_i - a_i)$. Therefore, a random number is generated in two steps. First, a random variable r uniformly distributed in interval $[0, 1]$ is generated. If r is larger than $(b_i - m_i(y))/(b_i - a_i)$, then $x = b_i$, i.e., the generated number at the second step is b_i. If r is smaller than $(b_i - m_i(y))/(b_i - a_i)$, then we use the well-known inverse transformation method. According to the method, the random number x is computed through the inverse lower CDF, i.e.,

$$x = \frac{m_i(y) - a_i \cdot r}{1 - r}.$$

The right side of the above equality is obtained by means of the inverse transformation of the lower CDF.

The same simulation procedure can be provided for the upper probability distribution. A random variable r uniformly distributed in interval $[0, 1]$ is generated. If r is smaller than $(b_i - m_i(y))/(b_i - a_i)$, then $x = a_i$, i.e., the generated number at the second step is a_i. If r is larger than $(b_i - m_i(y))/(b_i - a_i)$,

then, according to the inverse transformation method, the random number x is computed through the inverse upper CDF, i.e.,

$$x = \frac{m_i(y) - b_i \cdot (1 - r)}{r}.$$

6 Hinge loss function and SVM

A procedure for computing optimal values of parameters $\mathbf{w}$ depends on the loss function L. We consider the so-called hinge loss function which is of the form $L(f, y) = \max(0, 1 - yf)$. This function is taken for the consideration in order to reduce the classification problem to the SVM method which gives the opportunity to construct nonlinear classification models in a rather simple way.

After substituting the hinge loss function into the objective function (8), we get the following optimization problem:

$$\begin{aligned}\overline{R}(\mathbf{w}) = &\sum_{y=-1,1} \frac{P(y)}{n_y} \sum_{i \in N(y)} \frac{1}{K_i} \\ &\times \sum_{k=1}^{K_i} \max\left(0, 1 - y_i \left(\left\langle \left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)}\right), w\right\rangle + w_0\right)\right).\end{aligned}$$

It can be rewritten in a more dense form:

$$\begin{aligned}\overline{R}(\mathbf{w}) = &\sum_{i=1}^{n} \frac{P(y_i)}{n_{y_i} K_i} \\ &\times \sum_{k=1}^{K_i} \max\left(0, 1 - y_i \left(\left\langle \left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)}\right), w\right\rangle + w_0\right)\right).\end{aligned}$$

Let us introduce a new optimization variable

$$G_{i,k} = \max\left(0, 1 - y_i \left(\left\langle \left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)}\right), w\right\rangle + w_0\right)\right).$$

Then we get the optimization problem

$$\overline{R}(\mathbf{w}_{\text{opt}}) = \min_{w} \left(\sum_{i=1}^{n} \frac{P(y_i)}{n_{y_i} K_i} \sum_{k=1}^{K_i} G_{i,k}\right), \tag{9}$$

subject to

$$G_{i,k} \geq 1 - y_i \left(\left\langle \left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)}\right), w\right\rangle + w_0\right), \tag{10}$$

$$G_{i,k} \geq 0, \ \forall i = 1, ..., n, \ k = 1, ..., K_i. \tag{11}$$

So, we have the linear optimization problem having $(l + 1) + \sum_{i=1}^{n} K_i$ optimization variables and $2 \sum_{i=1}^{n} K_i$ constraints.

Let us add the standard Tikhonov regularization term $\frac{1}{2}\langle w, w\rangle$ (the most popular penalty or smoothness term) [29] to the objective function (9) and the constant "cost" parameter C. The smoothness (Tikhonov) term can be regarded as a constraint which enforces uniqueness by penalizing functions with wild oscillation and effectively restricting the space of admissible solutions. The detailed analysis of regularization methods can be found also in the work [8]. Then we get the following quadratic programming problem:

$$\overline{R}(\mathbf{w}_{\rm opt}) = \min\left(\frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{n}\frac{P(y_i)}{n_{y_i}K_i}\sum_{k=1}^{K_i}G_{i,k}\right), \tag{12}$$

subject to (10)-(11).

Instead of minimizing the primary objective function (12), a dual objective function, the so-called Lagrangian, can be formed of which the saddle point is the optimum. The Lagrangian is

$$\begin{aligned} L = &\frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{n}\sum_{k=1}^{K_i}\frac{P(y_i)}{n_{y_i}K_i}G_{i,k} - \sum_{i=1}^{n}\sum_{k=1}^{K_i}\eta_{i,k}G_{i,k} \\ &- \sum_{i=1}^{n}\sum_{k=1}^{K_i}\varphi_{i,k}\left(G_{i,k} - 1 + y_i\left\langle\left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)}\right), w\right\rangle + y_i w_0\right). \end{aligned}$$

Here $\eta_{i,k}, \varphi_{i,k}$, $i = 1, ..., n$, $k = 1, ..., K_i$, are Lagrange multipliers. Hence, the dual variables have to satisfy positivity constraints $\eta_{i,k} \geq 0, \varphi_{i,k} \geq 0$ for all i, k.

Hence, we get the simplified Lagrangian

$$L = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{n}\sum_{k=1}^{K_i}\varphi_{i,k}\left(y_i\left\langle\left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)}\right), w\right\rangle - 1\right).$$

Now we can divide all terms of the above objective function into two parts corresponding to the observed and unobserved features, respectively,

$$\begin{aligned} L = &\sum_{i=1}^{n}\sum_{k=1}^{K_i}\varphi_{i,k} + \frac{1}{2}\left\langle w^{(1)}, w^{(1)}\right\rangle \\ &- \sum_{i=1}^{n}\sum_{k=1}^{K_i}\varphi_{i,k}\left(y_i\left\langle\mathbf{x}_i^{(1)}, w^{(1)}\right\rangle\right) \\ &+ \frac{1}{2}\left\langle w^{(2)}, w^{(2)}\right\rangle - \sum_{i=1}^{n}\sum_{k=1}^{K_i}\varphi_{i,k}\left(y_i\left\langle\mathbf{x}_{i,k}^{(2)}, w^{(2)}\right\rangle\right). \end{aligned}$$

Hence, we obtain the dual optimization problem

$$\begin{aligned}\max_{\varphi_{i,k}} L = &\sum_{i=1}^{n}\sum_{k=1}^{K_i}\varphi_{i,k} \\ &- \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \left\langle \mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)} \right\rangle \sum_{k=1}^{K_i}\sum_{u=1}^{K_j}\varphi_{i,k}\varphi_{j,u} \\ &- \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \sum_{k=1}^{K_i}\sum_{u=1}^{K_j}\varphi_{i,k}\varphi_{j,u} \left\langle \mathbf{x}_{i,k}^{(2)}, \mathbf{x}_{j,u}^{(2)} \right\rangle,\end{aligned}$$

subject to

$$0 \le \varphi_{i,k} \le \frac{C \cdot P(y_i)}{n_{y_i} K_i}, \quad i = 1, ..., n, \ k = 1, ..., K_i.$$

Any data point for which $\varphi_{i,k} > 0$ is called a support vector. Let S and N_S denote the set of indices of the support vectors and their total number, respectively. Then one of the ways for computing the parameter w_0 is

$$w_0 = \frac{1}{N_S}\sum_{s\in S}\left(y_s - \sum_{i=1}^{n}\sum_{k=1}^{K_i} y_i \varphi_{i,k} \left\langle \left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)} \right), \left(\mathbf{x}_s^{(1)}, \mathbf{x}_s^{(2)} \right) \right\rangle \right),$$

where $(y_s, \mathbf{x}_s)$ is one of the support vectors.

If we assume that $K_i = K$ for all $i = 1, ..., n$, the prior probabilities are defined as $P(y) = n_y/n$, then we rewrite the optimization problem as

$$\begin{aligned}\max_{\varphi_{i,k}} L = &\sum_{i=1}^{n}\sum_{k=1}^{K}\varphi_{i,k} \\ &- \frac{1}{2}\sum_{i,j=1}^{n}\sum_{k,u=1}^{K} y_i y_j \left\langle \left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)} \right), \left(\mathbf{x}_j^{(1)}, \mathbf{x}_{j,u}^{(2)} \right) \right\rangle \varphi_{i,k}\varphi_{j,u},\end{aligned}$$

subject to

$$0 \le \varphi_{i,k} \le \frac{C}{nK}, \quad i = 1, ..., n, \ k = 1, ..., K.$$

Finally, we can write the discriminant function

$$f(\mathbf{x}) = \sum_{i=1}^{n}\sum_{k=1}^{K} y_i \varphi_{i,k} \left\langle \left(\mathbf{x}_i^{(1)}, \mathbf{x}_{i,k}^{(2)} \right), \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \right) \right\rangle + w_0.$$

The main advantage of the SVM is the use of kernels which are functions that transform the input data to a high-dimensional space where the learning problem is solved. There are many types of kernel that may be used in an SVM. Acceptable kernels must satisfy Mercer's condition. Commonly used forms of kernels are linear $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, polynomial $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \langle \mathbf{x}_i, \mathbf{x}_j \rangle + r)^d$,

$\gamma > 0$, radial basis function (RBF) $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$, $\gamma > 0$, sigmoid $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\gamma \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle + r\right)$. Here γ, r, and d are kernel parameters. The kernel functions allow us to significantly extend the class of discriminant functions that can be used in this approach.

7 Experimental design

We illustrate the method proposed in this paper via several examples, all computations have been performed using the statistical software R [22]. We investigate the performance of the proposed method and compare it with other methods dealing with missing data by considering the accuracy measure (ACC), which is the proportion of correctly classified cases on a sample of data, i.e., ACC is an estimate of a classifier's probability of a correct response. This measure is often used to quantify the predictive performance of classification methods and it is an important statistical measures of the performance of a binary classification test. It can formally be written as $ACC = N_T/N$. Here N_T is the number of test data for which the predicted class for an example coincides with its true class, and N is the total number of test data.

First we consider a numerical example with synthetic data. In this example, we generate instances with two features ($l = 2$) such that the second feature is unobserved. We generate 500 normally distributed random values for every features with the expectations $m_1(-1) = 4$, $m_1(1) = 6$, $m_2(-1) = 5$, $m_2(1) = 10$, and the standard deviations $\sigma_1 = 1$ and $\sigma_2 = 3$, respectively. We take identical standard deviations for both classes in order to simplify the example. Moreover, we state the lower and upper bounds for values of the second feature $a_2 = 4$ and $b_2 = 14$. Then we randomly select $n = 10$ points (instances) with identical numbers of points (5) for both classes and get three training sets. The first and the second training sets are obtained in the following way. We generate the values of the second feature $K = 20$ times for every example. In sum, we have 200 examples. At that, the values for the first training set are generated in accordance with the CDFs $\widetilde{F}_j(x_j \mid 1)$ for the class $y = 1$ and with the CDFs $\widetilde{F}_j(x_j \mid -1)$ for the class $y = -1$. This training set corresponds to the minimax strategy. The values for the second training set are generated in accordance with the CDFs $\widehat{F}_j(x_j \mid -1)$ for the class $y = -1$ and with the CDFs $\widehat{F}_j(x_j \mid 1)$ for the class $y = 1$. The second training set corresponds to the minimin strategy. For getting the third training set, we replace all values of the second feature in the set of $n = 10$ examples by the expectations $m_2(y)$ for $y = -1$ and $y = 1$. Here we use the available mean values of the second feature as values of the feature. We will call this strategy as direct for short. The initially generated 500 normally distributed random values will be used for testing resulting discriminant functions.

The ACC measures and the discriminant functions for the above three training sets will be indexed by numbers 1, 2, 3 corresponding to the minimax, minimin and direct strategies, respectively.

Table 1: The ACC measures for linear and RBF kernels by normally distributed values of features

	linear kernel			RBF kernel		
n	ACC1	ACC2	ACC3	ACC1	ACC2	ACC3
10	0.41	0.85	0.85	0.59	0.87	0.82

Table 2: The ACC measures for linear and RBF kernels by exponentially distributed values of the second feature

	linear kernel			RBF kernel		
n	ACC1	ACC2	ACC3	ACC1	ACC2	ACC3
10	0.63	0.61	0.59	0.65	0.65	0.62

We will use the linear and RBF kernels with the parameter $\gamma = 1/l$. By applying the above initial data, we get three discriminant functions corresponding to three strategies (minimax, minimin, direct):

$$f_1(\mathbf{x}) = -1.18x_1 + 0.445x_2 + 2.24,$$
$$f_2(\mathbf{x}) = 0.009x_1 - 0.49x_2 + 3.91.$$
$$f_3(\mathbf{x}) = -0.00004x_1 - 0.5x_2 + 4.$$

The corresponding ACCs for linear and RBF kernels are shown in Table 1. One can see from the table that the optimistic and direct strategies provide better results in comparison with the minimax strategy. This can be explained by exploiting the normal distribution (symmetric and unimodal) with rather small standard deviations for generating the random values of the second feature.

We replace the normal distribution of the second feature values by the truncated exponential distribution with the CDF $1 - \exp((x - a_2)/m_2(y))$ if $x < b_2$ and 1 if $x \geq b_2$. This distribution is not symmetric and its mean value can not replace the corresponding random values. By taking the linear and RBF kernels, $n = 10$, $K = 20$, we get the following discriminant functions:

$$f_1(\mathbf{x}) = -1.96x_1 + 0.2x_2 + 7.91,$$
$$f_2(\mathbf{x}) = -0.09x_1 - 0.47x_2 + 4.2,$$
$$f_3(\mathbf{x}) = -0.5x_2 + 4.$$

The corresponding ACCs for linear and RBF kernels are shown in Table 2. It can be seen from the table that the minimax strategy provides better results. It follows from the fact that the minimax strategy takes into account worst cases of the probability distribution of feature values. Of course, the exploited exponential distribution is not the worst case, but it is not the best case too. We can immediately observe that change for the worse of the probability distribution leads to improving the minimax strategy in comparison with the minimin and direct strategies.

The proposed method has been evaluated and investigated by the following publicly available datasets: Iris, Pima Indian Diabetes, Mammographic masses,

Table 3: A brief introduction about datasets

Dataset	N	l	n_{-1}	n_{+1}
Iris	150	4	50	150
Pima Indian Diabetes	768	8	268	500
Mammographic masses	961	4	445	516
Parkinsons	195	23	48	147
Breast Cancer Wisconsin (Original)	699	9	458	241
Breast Cancer Wisconsin (Diagnostic)	569	32	212	357
Musk	476	166	207	269
Indian Liver Patients	583	10	167	416
Lung-cancer	32	57	9	23

Parkinsons, Indian Liver Patient, Breast Cancer Wisconsin (Original), Breast Cancer Wisconsin (Diagnostic), Musk, Lung-cancer. All data sets are from the UCI Machine Learning Repository [13]. Table 3 is a brief introduction about these datasets, while more detailed information can be found from, respectively, the data resources.

For all data we use the repeated random sub-sampling validation procedure, i.e., we randomly split the dataset into two subsets. One of them (training set having n instances) is used to train the model while the other (test set having $N - n$ instances) is used to validate the model. The number of instances for training will be denoted as n. Moreover, we take $n/2$ instances from every class for training. They are randomly selected from the classes. The remaining instances in the dataset are used for validation. The parameter of the RBF kernel γ for every dataset is chosen in order to maximize the accuracy measure. It is carried out by means of the following procedure. It is well known that letting the C and γ grow exponentially is a practical method to identify good parameters. An $r \times r$ uniform grid in the logarithmic coordinate space ($C' = \log_2 C$, $\gamma' = \log_2 \gamma$) is usually used. The point in the grid represents a parameter pair (C', γ'). However, we fix the value of $C = 100$ in order to reduce the number of experiments because our main aim is to compare the proposed models with known models. So, we perform experiments on a 13 uniform grid where γ' has a range $2^{-6}, ..., 2^{6}$.

From every dataset, we randomly select a feature corresponding to missing values and compute its mean values for negative and positive labels, respectively. Moreover, we find the smallest and largest values of the selected feature which will be used for determining the lower and upper cumulative distribution functions. Then we generate the random values of the selected feature $K = 20$ times for every instance. In sum, we have nK instances. The above procedure is repeated $N = 50$ times such that the selected feature with missing values is chosen randomly in every iteration. In addition to the minimin (ACC1), minimax (ACC2) and direct (ACC3) strategies, we generate random values of the "missing" feature in accordance with the normal distribution and compute the corresponding accuracy measure ACC4. By using the RBF kernels and the

Table 4: The ACC measures for real datasets by different values of n

Dataset	n	ACC1	ACC2	ACC3	ACC4
Iris	20	0.961	0.982	0.977	0.998
	40	0.944	0.963	0.986	0.998
Pima	20	0.501	0.598	0.502	0.573
Indian Diabetes	40	0.532	0.655	0.590	0.621
Mammographic	20	0.642	0.769	0.750	0.722
masses	40	0.654	0.781	0.778	0.739
Parkinsons	20	0.567	0.721	0.706	0.637
	40	0.56	0.721	0.713	0.644
Breast Cancer	20	0.885	0.923	0.937	0.963
Wisconsin (Original)	40	0.832	0.919	0.815	0.962
Breast Cancer	20	0.820	0.835	0.885	0.921
Wisconsin (Diagnostic)	40	0.785	0.851	0.825	0.926
Musk	20	0.565	0.648	0.625	0.629
	40	0.606	0.678	0.689	0.658
Indian Liver	20	0.508	0.674	0.666	0.614
Patients	40	0.519	0.682	0.660	0.621
Lung-cancer	12	0.647	0.709	0.693	0.637
	16	0.666	0.679	0.637	0.786

cost parameter $C = 100$, we get the ACC measures for different values of n, whose values are shown in Table 4. These measures are mean values of the corresponding ACCs computed for every iteration.

One can see from Table 4 that the proposed minimax strategy (ACC2) outperforms the direct strategy and the normal distribution imputation procedure for some real datasets. Of course, there are datasets for which the measures ACC3 or ACC4 are larger than ACC2. If we have seen from the experiments with synthetic data that the minimax strategy provides better results when the distribution of the feature values is not symmetric and its mean value can not replace the corresponding random values, then it is difficult to determine clear conditions of using the proposed model with real data. We can say that these conditions directly depend on a probability distribution of the feature values in real data. When we do not have this information, the proposed method should be used jointly with other models dealing with missing data.

8 Conclusion

A classification problem under partial information about some features in the form of conditional expectations or mean values of features for every class has been studied in the paper. Its solution is based on the pessimistic (minimax) and optimistic (minimin) decision strategies.

What are the main advantages of the proposed method? First, the classifi-

cation algorithm totally exploits the available information in the form of mean values of some features and the bounds of these features. At the same time, it does not employ any additional information which may be unjustified and incorrect. It does not use also additional assumption which may lead to incorrect prediction results. Second, the proposed method has a strong probabilistic background, and this fact allows us to use it in arbitrary applications where the initial information is scarce. Third, the method exploits the well-known minimax and minimin strategies which have a strong explanation. A cautious decision strategy as an intermediate case between pessimistic and optimistic strategies with a predefined caution parameter can also be studied in the same way. However, this is a direction for further research. Fourth, the method is reduced to the SVM. This fact allows us to simply construct non-linear classification models by using suitable kernels. Fifth, the method allows us to reduce the classification problem to the standard form. This implies that a standard software can be applied for its implementation. The algorithm for computing the optimal parameters of every classification model can be easily implemented with standard functions of the statistical software package R or by using the well-known software library LIBSVM (A Library for Support Vector Machines) ([5]).

The numerical examples have illustrated that the minimax classifiers can provide more accurate results in many cases in spite of their over-conservative decisions. At the same time, the given experiments can be viewed as a preliminary study of the proposed framework for applying the imprecise models to classification problems with missing values. An additional study has to be carried out in order to totally figure out when the proposed classifiers outperform the available classification models.

One can also see from the paper that the Monte Carlo technique is a versatile tool for dealing with partial information. Various classification problems under different types of partial and unreliable information could be solved in the same way. A detailed analysis of the corresponding classification models is another direction for further research.

At the same time, it is well known that one possible limitation of Monte-Carlo methods is the strong dependence of computational effort (proportional to the number of samplings). This implies that the learning of large datasets may lead to a hard computational problem. However, first of all, the minimax strategy should be used when the number of instances in training sets is rather small in order to provide the robust classification. When the training set consists of a large number of instances other models might give better results. Second, variance reduction techniques can be applied to the classification procedures to decrease the computational effort. This is also a topic of further research.

The proposed method can be also extended on the case of interval-valued mean values of unobserved features. In this case, the lower and upper CDFs are determined by the lower and upper mean values of features.

Acknowledgement

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

References

[1] R. Alaiz-Rodríguez, A. Guerrero-Curieses, and J. Cid-Sueiro. Minimax regret classifier for imprecise class distributions. *J. Mach. Learn. Res.*, 8:103–130, 2007.

[2] R. Alaiz-Rodríguez, A. Guerrero-Curieses, and J. Cid-Sueiro. Improving classification under changes in class and within-class distributions. In J. Cabestany, F. Sandoval, A. Prieto, and J. Corchado, editors, *Bio-Inspired Systems: Computational and Ambient Intelligence*, volume 5517 of *Lecture Notes in Computer Science*, pages 122–130. Springer Berlin / Heidelberg, 2009.

[3] G.E.A.P.A. Batista and M.C. Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003.

[4] J.O. Berger and G. Salinetti. Approximations of Bayes decision problems: the epigraphical approach. *Annals of Operations Research*, 56:1–13, 1995.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[6] S. Destercke, D. Dubois, and E. Chojnacki. Unifying practical uncertainty representations - i: Generalized p-boxes. *International Journal of Approximate Reasoning*, 49(3):649–663, 2008.

[7] R. Esposito and L. Saitta. Monte Carlo theory as an explanation of bagging and boosting. In *Procedings of IJCAI'03*, pages 499 – 504, 2003.

[8] T. Evgeniou, T. Poggio, M. Pontil, and A. Verri. Regularization and statistical learning theory for data analysis. *Computational Statistics & Data Analysis*, 38(4):421 – 432, 2002.

[9] A. Farhangfar, L. Kurgan, and J. Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41(12):3692–3705, 2008.

[10] A. Farhangfar, L. Kurgan, and J. Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41:3692–3705, 2008.

[11] S. Ferson, L. Ginzburg, and R. Akcakaya. Whereof one cannot speak: When input distributions are unknown. Technical report, Applied Biomathematics Report, 2001. http://www.ramas.com/whereof.pdf.

[12] S. Ferson, V. Kreinovich, L. Ginzburg, D.S. Myers, and K. Sentz. Constructing probability boxes and Dempster-Shafer structures. Report SAND2002-4015, Sandia National Laboratories, January 2003.

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

[14] S. Garcia and F. Herrera. An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[15] J. Grzymala-Busse and M. Hu. A comparison of several approaches to missing attribute values in data mining. In *Rough sets and current trends in computing*, pages 378–385. Springer, Berlin/Heidelberg, 2001.

[16] J.E. Hurtado. An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory. *Structural Safety*, 26(3):271 – 293, 2004.

[17] J.E. Hurtado and D.A. Alvarez. Classification approach for reliability analysis with stochastic finite-element modeling. *Journal of Structural Engineering*, 129(8):1141–1149, 2003.

[18] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[19] V. P. Kuznetsov. *Interval Statistical Models*. Radio and Communication, Moscow, 1991. in Russian.

[20] J. Luengo, S. Garcia, and F. Herrera. On the choice of the best imputation methods for missing values considering three groups of classification methods. *Knowledge and information systems*, 32(1):77–108, 2012.

[21] Jianhui Ning and P.E. Cheng. A comparison study of nonparametric imputation methods. *Statistics and Computing*, 22(1):273–285, 2012.

[22] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005.

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

[24] D.B. Rubin. Multiple imputation after 18+ years. *Journal of the Americal Statistical Association*, 91(434):473–489, June 1996.

[25] M. Saar-Tsechansky and F. Provost. Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8:1625–1657, 2007.

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

[27] J. Shao. Monte Carlo approximations in Bayesian decision theory. *Journal of the American Statistical Association*, 84:727–732, 1989.

[28] P. Sollich. Bayesian methods for Support Vector Machines: Evidence and predictive class probabilities. *Machine Learning*, 46:21–52, 2002.

[29] A.N. Tikhonov and V.Y. Arsenin. *Solution of Ill-Posed Problems*. W.H. Winston, Washington DC, 1977.

[30] L.V. Utkin. Regression analysis using the imprecise Bayesian normal model. *Int. J. Data Analysis Techniques and Strategies*, 2(4):356–372, 2010.

[31] L.V. Utkin and F.P.A. Coolen. On reliability growth models using Kolmogorov-Smirnov bounds. *International Journal of Performability Engineering*, 7(1):5–19, 2011.

[32] L.V. Utkin and Y.A. Zhuk. A machine learning algorithm for classification under extremely scarce information. *International Journal of Data Analysis Techniques and Strategies*, 4(2):115–133, 2012.

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

[34] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

[35] P. Walley. Measures of uncertainty in expert systems. *Artificial Intelligence*, 83:1–58, 1996.

[36] G.M. Weiss. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6:7–19, 2004.

[37] D. Williams, X. Liao, Y. Xue, L. Carin, and B. Krishnapuram. On classification with incomplete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:427–436, 2007.