

Adaptive Robust SVM-Based Classification Algorithms for Multi-Robot Systems using Sets of Weights

Lev V. Utkin Vladimir S. Zaborovsky Sergey G. Popov

Telematics Department, Peter the Great St.Petersburg Polytechnic University, Russia
lev.utkin@gmail.com, vlad@neva.ru, popovserge@spbstu.ru

Abstract

Three adaptive iterative minimax multi-robot system learning algorithms are proposed under condition that every observation obtained from robots is set-valued, i.e., it consists of several elements. The set-valued data are caused due to a fact that robots in the system provide different measurements in a single system observation. The first idea underlying the algorithms is to use sets of weights or imprecise weights of a special form for all elements of training data. The second idea is to apply the imprecise Dirichlet model for iterative updating the sets of weights depending on the classification accuracy and for assigning new weights to robots for improving classifiers. The simplest first algorithm is a modification of the SVM in order to take into account set-valued data. The second algorithm is the AdaBoost with the modified SVM under set-valued data. The third algorithm is the modification of the AdaBoost with updating imprecise weights of robots. The algorithms allow us to take into account the set-valued observations in the framework of the minimax decision strategy and to get optimal weights of robots to improve the classification accuracy of the trained multi-robot system.

Keywords: multi-robot system, SVM, classification, AdaBoost, set-valued observations, sets of weights

1 Introduction

Multi-Robot Systems (MRS) have drawn increasing attention last time due to several factors including the ability to perform complex tasks more efficiently compared to single-robot systems (Navarro and Matia, 2013; Tan and Zheng, 2013). In the MRS, robots are usually equipped with several sensors used to acquire as much information about the external world as possible. It is pointed out in (Pronobis et al., 2008) that each sensor may capture a different aspect of the environment, however, alternative interpretations of the information obtained by the same sensor can also be valuable. One of the important problems of the MRS learning is to integrate effectively multiple distributed sensor suites which may include GPS units, temperature sensors, altimeters, imaging systems, etc. (Cowley et al., 2004).

There are a lot of approaches for combining the multi-sensor information in robotics system learning (see, for

example, (Du et al., 2012; Khamis et al., 2015; Ravet et al., 2013; Yuksel et al., 2012)) which mainly exploit various weighting schemes (in terms of probabilities or other measures) to differentiate the learning data sources and combine them in accordance with predefined rules taking into account the quality or reliability of data from different sensors. However, the most approaches assume that there is a large training set for learning and for assigning weights to sensors. This assumption may be violated in many applications, especially, during an initial learning phase when it is very difficult to estimate every robot or its sensors in order to apply the available weighted schemes.

Two main strategies of MRS learning can be marked out. The first strategy is when robots perform cooperative classification based on the same meta-classifier trained from training data obtained from sensors and from the feedback provided by a human teacher. This strategy may be useful during an initial learning phase when we do not know how different robots behave in their team and how reliable the sensor information provided by each robot. The second strategy is when each robot trains its own classifier (Di Caro et al., 2013), using the features extracted from a set of a locally available labeled examples, which correspond to the sensor information acquired from the robots' specific viewpoint and exploiting the experience from other robots.

The first strategy is studied in the present paper. We consider a case when it is difficult or just impossible to assign weights to separating sensors in order to combine their suites by using weighting schemes. The main difficulty in the joint use of training data from several sensors during the initial learning phase is that we cannot consider every sensor data as a separate training example. Let us consider for example the temperature sensors which provide with the environment temperature from a set of robots at some time moment. Every sensor provides with the information about the temperature of the same object approximately at the same time. Therefore, the set of temperature measurements in this case should be regarded as a single multi-viewed training example. Of course, we can use, for instance, some middle temperature for training. However, this combination rule says that all robots are identically reliable and accurate. This assumption is too strong in order to be valid in many applications. The initial learning phase is characterized by the lack of the

corresponding knowledge. Therefore, we propose a learning algorithm which takes into account the above peculiarities. It should be noted that only the initial learning phase may be available in some applications, and the learning algorithm proposed for this phase is entirely used in the MRS learning.

One of the most efficient and popular methods for the MRS classification learning is the support vector machine (SVM). Another efficient method is the AdaBoost proposed in (Freund and Schapire, 1997). Therefore, we propose a modification of the AdaBoost with SVMs of a special form as weak learners, which takes into account the fact that training data in the MRS are obtained from a set of unknown robots or their sensors. Moreover, we modify also SVMs such that imprecise judgements about robots can be incorporated into the SVM in order to improve the classification performance of the MRS. The main idea underlying the proposed modifications is the following. Training data from every robot are viewed as training examples, but their weights are replaced by some sets of weights such that every training example can be regarded as a set-valued data. Then we apply the robust minimax strategy in order to find an optimal decision function separating set-valued observations from different classes. The sets of weights are derived from the imprecise available information about robots. Moreover, we propose a double adaptive algorithm. The first adaptation is performed by the AdaBoost through changes of observation weights. The second adaptation is updating of the weight sets of set-valued observations in accordance with a number of correctly classified measurements from every robot at every iteration of the AdaBoost. The second adaptation is implemented by means of the imprecise Dirichlet model (Walley, 1996). In fact, we propose three algorithms for learning the MRS. The simplest one is just a modification of the SVM in order to take into account set-valued data. The second algorithm is the AdaBoost with the modified SVM under set-valued data. The third algorithm is the modification of the AdaBoost with updating imprecise weights of robots. The complexity of the algorithms do not differ from the complexity of the corresponding standard SVM and AdaBoost algorithms.

2 Formal Problem Statement and SVM

Suppose that we have observations or measurements from all sensors of T robots at every time moment k . After time moment n , we get the training set $S = \{(\mathbf{A}_1, y_1), \dots, (\mathbf{A}_n, y_n)\}$, where \mathbf{A}_k is a matrix having T rows $\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_T^{(k)}$, $k = 1, \dots, n$, and m columns such that the row $\mathbf{x}_j^{(k)}$ is a vector of all measurements (features) obtained from the j -th robot. We assume that there are two classes (the binary classification) and $y_i \in \{-1, 1\}$. The learning aim is to construct an accurate classifier $c: \mathbb{R}^{m \cdot T} \rightarrow \{-1, 1\}$ that maximizes the probability that

$$c(\mathbf{A}_i) = y_i \text{ for } i = 1, \dots, n.$$

One of the ways for classification is to find a real valued separating function $f(\mathbf{x}, \mathbf{w}, b)$ having parameters \mathbf{w} and b such that $\mathbf{w} = (w_1, \dots, w_m) \in \mathbb{R}^m$ and $b \in \mathbb{R}$, for example, $f(\mathbf{x}, \mathbf{w}, b) = \langle \mathbf{w}, \mathbf{x} \rangle + b$. Here $\langle \mathbf{w}, \mathbf{x} \rangle$ denotes the dot product of two vectors \mathbf{w} and \mathbf{x} . We also denote $w = \langle \mathbf{w}, b \rangle$. We assume for simplicity that, after the learning phase, every robot uses the separating function $f(\mathbf{x}, \mathbf{w}, b)$. Though we can also apply the function $f(\mathbf{A}, \mathbf{w}, b)$ which is defined for matrix \mathbf{A} in the case of the inter-robot transfer learning.

One of the simplest ways for solving the classification problem is to replace every column of \mathbf{A}_k by a number, for example, by the mean value of all elements of the column, and to apply the standard SVM.

In order to give a short description of the well-known SVM, we replace the set S by a set $S^* = \{(\mathbf{x}_1^*, y_1), \dots, (\mathbf{x}_n^*, y_n)\}$. Here \mathbf{x}_i^* is the vector of replaced values for every feature. Let ϕ be a feature map $\mathbb{R}^m \rightarrow G$ such that the data points are mapped into an alternative higher-dimensional feature space G . In other words, this is a map into an inner product space G such that the inner product in the image of ϕ can be computed by evaluating some simple kernel $K(\mathbf{x}_i^*, \mathbf{x}_j^*) = (\phi(\mathbf{x}_i^*), \phi(\mathbf{x}_j^*))$ such as the Gaussian kernel. The SVM minimizes the empirical risk measure with a smoothness or penalty term $\langle \mathbf{w}, \mathbf{w} \rangle / 2$:

$$R(w) = \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^n l(y_i, \mathbf{x}_i^*, w). \quad (1)$$

Here C is the tuning ‘‘cost’’ parameter C which balances the trade-off between the empirical risk measure and the penalty term (Scholkopf and Smola, 2002); $l(y_i, \mathbf{x}_i^*, w)$ is the classification loss function. The so-called hinge loss function is used in SVM, i.e., $l(y, \mathbf{x}, w) = \max(0, 1 - y \cdot f(w, \phi(\mathbf{x})))$. Hence, the SVM classifier can be represented in the form of the following convex optimization problem with slack variables ξ_i , $i = 1, \dots, n$:

$$\min_{\xi, w} R(w) = \min_{\xi, w} \left(\frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^n \xi_i \right) \quad (2)$$

subject to

$$\xi_i \geq 0, \quad y_i (\langle w, \phi(\mathbf{x}_i^*) \rangle + b) \geq 1 - \xi_i, \quad i = 1, \dots, n. \quad (3)$$

The quantity $C\xi_i$ is the ‘‘penalty’’ for any data point \mathbf{x}_i^* that either lies within the margin on the correct side of the hyperplane ($\xi_i \leq 1$) or on the wrong side of the hyperplane ($\xi_i > 1$).

Instead of minimizing the primary objective function (2) with constraints (3), we use a dual programming problem:

$$\max_{\alpha} \left(\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i^*, \mathbf{x}_j^*) \right) \quad (4)$$

subject to

$$\sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n. \quad (5)$$

Here $\alpha_i, i = 1, \dots, n$, are Lagrange multipliers or optimization variables in (4)-(5). After substituting the obtained solution into the expression for the decision function f , we get the “dual” separating function

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i^*, \mathbf{x}) + b. \quad (6)$$

The parameter b is defined by using support vectors \mathbf{x}_i^* from the following equation

$$b = y_j - \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i^*, \mathbf{x}_j^*). \quad (7)$$

3 SVM by Set-Valued Training Data

The above approach for dealing with the training set S by replacing it with S^* cannot be used in the high-noise regime and by a rather small training set when robots provide scattered measurements. Therefore, in order to develop a robust classification procedure, we propose another approach for dealing with the training set S .

Let us consider a set of empirical expected risk measures $R(w)$ such that every measure from the set corresponds to a row, say $\mathbf{x}_j^{(k)}$, of the matrix \mathbf{A}_k . Then there exists an upper bound for $R(w)$, which is defined as

$$\bar{R}(w) = \max_{\mathbf{x}_i^{(k)} \in \mathbf{A}_k, k=1, \dots, n} \sum_{i=1}^n l(y_i, \mathbf{x}_i^{(k)}, w). \quad (8)$$

Here the expected risk is maximized over all $\mathbf{x}_i^{(k)}$ from $\mathbf{A}_k, k = 1, \dots, n$. The upper bound $\bar{R}(w)$ corresponds to the robust or pessimistic strategy in the sense that we select the “worst” elements $\mathbf{x}_0^{(k)}$ from \mathbf{A}_k .

Suppose that there are rows $\mathbf{x}_0^{(k)} \in \mathbf{A}_k$ for all $k = 1, \dots, n$, which provide the largest value of the expected risk. Then we can assign non-zero weights to the rows such that weights of other vectors $\mathbf{x}_i^{(k)} \neq \mathbf{x}_0^{(k)}$ become to be zero. This implies that the problem of maximization of the expected risk over rows of $\mathbf{A}_1, \dots, \mathbf{A}_n$ can be transformed to a problem of maximization of the expected risk over a set of weights. This transformation can be regarded as the uncertainty trick, i.e., we transform training data with the uncertainty of robot measurements to training data with the weight or probabilistic uncertainty.

Therefore, we extend the training set by rows $\mathbf{x}_i^{(k)}$ such that the extended training set has now $N = T \cdot n$ elements, but these elements have different weights. Let us denote a vector of new weights as $\pi = (\pi_1, \dots, \pi_N)$. Introduce also a set of indices $I_k = \{1 + (k-1)T, \dots, T + (k-1)T\}$. We only know about π that the sum of weights of all rows

from \mathbf{A}_k is $\sum_{i \in I_k} \pi_i = 1/n$ because every element of the initial training set has the weight or the probability $1/n$. This implies that the set \mathcal{P} produced by all possible distributions π is convex, and there is an upper bound for $R(w)$, which is written as

$$\bar{R}(w) = \max_{\pi \in \mathcal{P}} \sum_{k=1}^n \sum_{i \in I_k} \pi_i l(y_i, \mathbf{x}_i^{(k)}, w). \quad (9)$$

It is important to point out that we did not simply extend the training set. By adding new elements to the training set, we change weights of the elements. At that, the weights of new elements are only partly known, and they belong to the set \mathcal{P} .

Now we can construct a modification of SVM taking into account the robust strategy, which is formulated as the following minimax optimization problem:

$$\min_w \bar{R}(w) = \min_w \max_{\pi \in \mathcal{P}} R(w). \quad (10)$$

Let us fix variables w and consider only a problem with variables $\pi \in \mathcal{P}$ by fixed w . The upper bound for $R(w)$ can be found by solving the optimization problem:

$$\bar{R}(w) = \max_{\pi \in \mathcal{P}} \sum_{k=1}^n \sum_{i \in I_k} \pi_i l(y_i, \mathbf{x}_i^{(k)}, w) \quad (11)$$

subject to

$$\sum_{i \in I_k} \pi_i = \frac{1}{n}, \quad k = 1, \dots, n, \quad \sum_{k=1}^n \sum_{i \in I_k} \pi_i = 1. \quad (12)$$

The above constraints stem from the weights $1/n$ of initial training elements and from the sum of weights of all rows. It should be noted that the above optimization problem is linear and the following dual optimization problem can be written:

$$\bar{R}(w) = \min \left\{ c_0 + \frac{1}{n} \sum_{k=1}^n c_k \right\} \quad (13)$$

subject to $c_0, c_k \in \mathbb{R}, k = 1, \dots, n$,

$$c_0 + \sum_{k=1}^n c_k \mathbf{1}(i \in I_k) \geq l(y_i, \mathbf{x}_i^{(k)}, w), \quad i = 1, \dots, N. \quad (14)$$

Here c_0, c_k are new optimization variables; $\mathbf{1}(D)$ is the indicator function taking the value 1 if D is true. If we assume that sensor measurements are different for every training example, then the last constraints can be simplified as

$$c_0 + c_k \geq \max_{i \in I_k} l(y_i, \mathbf{x}_i^{(k)}, w). \quad (15)$$

Substituting the above constraint into the objective function, we get the upper expected risk

$$\bar{R}(w) = \min \left\{ \sum_{k=1}^n \max_{i \in I_k} l(y_i, \mathbf{x}_i^{(k)}, w) \right\}. \quad (16)$$

Substituting the hinge loss function into the objective function, adding the standard Tikhonov regularization term in order to restrict the class of admissible solutions and simplifying the problem, we get

$$\bar{R}(w) = \min \left(\frac{1}{2} \langle w, w \rangle + C \cdot \sum_{k=1}^n \xi_k \right) \quad (17)$$

subject to

$$\xi_k \geq 1 - y_k \cdot f(\phi(\mathbf{x}_i^{(k)}), w), \quad (18)$$

$$i \in I_k, \xi_k \geq 0, k = 1, \dots, n. \quad (19)$$

The corresponding dual optimization problem (the Lagrangian) with variables α_i can be simply obtained

$$\max \left(-\frac{1}{2} \sum_{k=1}^n \sum_{t=1}^n \sum_{i \in I_k} \sum_{j \in I_t} \alpha_i \alpha_j y_k y_t K(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(t)}) + \sum_{k=1}^n \sum_{i \in I_k} \alpha_i \right) \quad (20)$$

subject to

$$\sum_{k=1}^n \sum_{i \in I_k} \alpha_i y_i = 0, \quad (21)$$

$$0 \leq \sum_{i \in I_k} \alpha_i \leq C, \alpha_i \geq 0, i \in I_k, k = 1, \dots, n. \quad (22)$$

If we compare the above optimization problem with the standard SVM, then we can see that variables α_i are restricted in a different way (see constraints (22)).

If we assume that all points of intervals produced by using a grid are different, i.e., they are unique for every interval, then the objective function (20) can be rewritten as

$$\max \left(-\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}) + \sum_{i=1}^N \alpha_i \right). \quad (23)$$

The constraint (21) can be rewritten in the same way

$$\sum_{i=1}^N \alpha_i y_i = 0, \alpha_i \geq 0, i \in I_k, \quad (24)$$

$$0 \leq \sum_{i \in I_k} \alpha_i \leq C, k = 1, \dots, n. \quad (25)$$

The separating function is of the form:

$$f(\mathbf{x}) = \sum_{k=1}^n \sum_{i \in I_k} \alpha_i y_i K(\mathbf{x}_i^*, \mathbf{x}) + b. \quad (26)$$

It can be seen from the above that the obtained SVM does not differ from the standard SVM with N training elements except for the last constraints where Lagrange multipliers are grouped in accordance with the robot information. When we have one robot, then $I_k = \{k\}$, $N = n$, and we get the standard SVM considered in the previous section.

4 A Modification of the AdaBoost

One of the efficient learning algorithms is the AdaBoost (Freund and Schapire, 1997). However, it is used for precise observations when the training set consists of point-valued examples. In order to improve the classification performance of the MRS, we propose a modification of the AdaBoost algorithm for the case of set-valued observations.

AdaBoost is a general purpose boosting algorithm that can be used in conjunction with many other learning algorithms to improve their performance via an iterative process. According to the AdaBoost, identical weights $h = (1/n, \dots, 1/n)$ are initially assigned to all examples. In each iteration, the weights of all misclassified examples are increased while the weights of correctly classified examples are decreased (see Algorithm 1). As a consequence, the weak classifier is forced to focus on the difficult examples of the training set by performing additional iterations and creating more classifiers. Furthermore, a weight ϕ_t is assigned to every individual classifier. This weight measures the overall accuracy of the classifier. Higher weights are assigned to more accurate classifiers. The weight distribution $h(t)$ is updated using the rule shown in Algorithm 1. The effect of this updating rule is to increase weights of misclassified examples and to decrease weights of correctly classified examples. Thus, the weights tend to concentrate on ‘‘hard’’ examples. The final classifier c is a weighted majority vote of T weak classifiers. We assume below that classifier c_t is the weighted SVM.

Algorithm 1 The AdaBoost algorithm

Require: Q (number of iterations), S (training set)

Ensure: $c_t, \phi_t, t = 1, \dots, Q$

- 1: $t \leftarrow 1; h_i(1) \leftarrow 1/n; i = 1, \dots, n$
 - 2: **repeat**
 - 3: Build classifier c_t using weights $h(t)$
 - 4: $e(t) \leftarrow \sum_{i: c_t(x_i) \neq y_i} h_i(t)$
 - 5: **if** $e(t) > 0.5$ **then**
 - 6: $Q \leftarrow t - 1$
 - 7: **exit** Loop
 - 8: **end if**
 - 9: $\phi_t \leftarrow \frac{1}{2} \ln \left(\frac{1-e(t)}{e(t)} \right)$
 - 10: $h_i(t+1) \leftarrow h_i(t) \cdot \exp(-\phi_t y_i c_t(x_i))$
 - 11: $t \leftarrow t + 1$
 - 12: **until** $t > Q$
 - 13: $c(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^Q \phi_i c_t(\mathbf{x}) \right)$
-

In order to modify the AdaBoost, we first provide a weighted version of the problem (20)-(22), namely, we suppose now the following condition for weights of observations:

$$\sum_{i \in I_k} \pi_i = h_k, k = 1, \dots, n, \sum_{k=1}^n h_k = 1. \quad (27)$$

The dual problem for computing $\bar{R}(w)$ in this case is

$$\bar{R}(w) = \min \left(c_0 + \sum_{k=1}^n c_k h_k \right) \quad (28)$$

subject to $c_0, c_k \in \mathbb{R}, k = 1, \dots, n$, and (14).

It is simply to prove that the primal optimization problem for minimizing the upper expected risk is

$$\bar{R}(w) = \min \left(\frac{1}{2} \langle w, w \rangle + C \cdot \sum_{k=1}^n h_k \xi_k \right) \quad (29)$$

subject to (19).

The corresponding dual optimization problem with variables α_i differs from (20)-(22) only by constraints

$$0 \leq \sum_{i \in I_k} \alpha_i \leq h_k C, \quad \alpha_i \geq 0, \quad i \in I_k, \quad k = 1, \dots, n, \quad (30)$$

where the upper bound for α_i is determined now by the weight h_k .

In order to use the AdaBoost, we define how to make decision about a class of a set-valued observation. A reasonable way is to apply one of the most popular strategy. According to the strategy, the set \mathbf{A}_k belongs to a class y if at least a half of its elements $\mathbf{x}_j^{(k)}$ belong to the class y , i.e., there holds

$$y_k^* = \arg \max_{y \in \{-1, 1\}} \sum_{i \in I_k} \mathbf{1}(c(\mathbf{x}_i^{(k)}) = y). \quad (31)$$

It is important to note that the proposed boosting algorithm deals with the extended training set consisting of N elements, but weights are modified only for set-valued observations, i.e., there are no restrictions for weights of elements from every set-valued observation except for the restriction (27).

5 Imprecise Updating Weights of Robots

So far we have considered the ‘‘worst’’ pessimistic case when we assumed almost total ignorance about the weight distribution over the robot measurements, i.e., we have assumed for the k -th local set of weights denoted as \mathcal{P}_k the restriction $\sum_{i \in I_k} \pi_i = h_k$. Now we propose an adaptive algorithm for reducing the local sets of weights and for incorporating an additional information about weights of robots, which is defined by the classification errors at every iteration of learning. A main idea underlying the adaptive algorithm is to apply the imprecise Dirichlet model (IDM) proposed in (Walley, 1996) for updating local sets of weights at every step. This idea is close to an algorithm proposed in (Utkin, 2015), which uses the IDM in the AdaBoost in order to avoid a problem of overfitting.

Suppose that, after constructing a classifier on the basis of n set-valued observations at the t -th iteration of the

boosting, we have $r_i^{(t)}$ correctly classified and $n - r_i^{(t)}$ misclassified measurements from the i -th robot, $i = 1, \dots, T$. This information allows us to update the local sets. It is reasonable to assume that if there are many misclassified measurements, then the set of weights $\mathcal{P}_k^{(t)}$ should be increased in order to make a robust decision. Moreover, it should be increased for robots whose measurements are misclassified.

In order to use the above information, we briefly consider the IDM. Let $U = \{u_1, \dots, u_T\}$ be a set of possible outcomes u_j . Assume the standard multinomial model: n observations are independently chosen from U with an identical probability distribution $\Pr\{u_j\} = p_j$ for $j = 1, \dots, T$, where each $p_j \geq 0$ and $\sum_{j=1}^T p_j = 1$. Then the IDM is defined in (Walley, 1996) as the set of all Dirichlet distributions over probabilities p_1, \dots, p_T whose parameters are s and mean values $\mathbf{q} = (q_1, \dots, q_T)$ such that \mathbf{q} belongs to the T -dimensional unit simplex denoted as $S(1, T)$. The hyperparameter s determines how quickly upper and lower probabilities of events converge as statistical data accumulate. Smaller values of s produce faster convergence and stronger conclusions, whereas large values of s produce more cautious inferences.

We propose to consider correctly classified measurements of the i -th robot as possible outcomes u_i . Then, according to Walley’s IDM, we can write the bounds for the probability of $r_i^{(t)}$ correctly classified measurements as follows:

$$\frac{r_i^{(t)}}{D^{(t)} + s} \leq p_i \leq \frac{r_i^{(t)} + s}{D^{(t)} + s}, \quad i = 1, \dots, T, \quad (32)$$

where $D^{(t)} = \sum_{i=1}^T r_i^{(t)}$ is the total number of correctly classified measurements at the t -th iteration.

Note that we have $r_i^{(t)} = D^{(t)} = 0$ before iterations of the boosting. Hence, $0 \leq p_i \leq 1$. If we multiply the bounds on $h_k(t)$, then we get bounds for probabilities from the set $\mathcal{P}_k^{(t)}$, i.e., there holds

$$\frac{r_i^{(t)} h_k(t)}{D^{(t)} + s} \leq \pi_{ki}^{(t)} \leq \frac{(r_i^{(t)} + s) h_k(t)}{D^{(t)} + s}, \quad i = 1, \dots, T. \quad (33)$$

Here $\pi_{ki}^{(t)}$ is the weight of the i -th robot measurement in the k -th observation at the t -th iteration such that

$$\sum_{i=1}^T \pi_{ki}^{(t)} = h_k(t), \quad k = 1, \dots, n. \quad (34)$$

Denote for simplicity

$$G_i = \frac{(r_i^{(t)} + s)}{D^{(t)} + s}, \quad F_i = \frac{r_i^{(t)}}{D^{(t)} + s}, \quad i = 1, \dots, T. \quad (35)$$

Then the dual optimization problem for computing $\bar{R}(w)$ at the t -th iteration is

$$\min \left(c_0 + \sum_{k=1}^n h_k(t) \left(c_k + \sum_{i=1}^T (g_{ki} G_i - d_{ki} F_i) \right) \right) \quad (36)$$

subject to $c_0, c_k \in \mathbb{R}, g_{kj} \geq 0, d_{kj} \geq 0, j = 1, \dots, T, k = 1, \dots, n$, and

$$c_0 + \sum_{k=1}^n c_k \mathbf{1}(i \in I_k) + \sum_{k=1}^n \sum_{j=1}^T (g_{kj} - d_{kj}) \mathbf{1}(I_k(j) = i) \geq l(y_i, \mathbf{x}_i^{(k)}, w), i = 1, \dots, N. \quad (37)$$

Here $I_k(j)$ is the j -th element of I_k . Let us write the Lagrangian by assuming that $l(y_i, \mathbf{x}_i^{(k)}, w)$ is the hinge loss function. It is of the form:

$$L = \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \cdot c_0 + C \sum_{k=1}^n c_k h_k(t) - \sum_{k=1}^n \sum_{i=1}^T g_{ki} \lambda_i - \sum_{k=1}^n \sum_{i=1}^T d_{ki} \mu_i + C \sum_{k=1}^n h_k(t) \sum_{i=1}^T (g_{ki} G_i - d_{ki} F_i) - \sum_{k=1}^n \sum_{i=1}^T (\beta_{ki} + \alpha_{ki}) (c_0 + c_k + g_{ki} - d_{ki}) - \sum_{k=1}^n \sum_{i=1}^T \alpha_{ki} (1 - y_k f(\phi(\mathbf{x}_i^{(k)}), w)). \quad (38)$$

Here $\alpha_{ki}, \mu_i, \lambda_i, i = 1, \dots, T, k = 1, \dots, n$, are Lagrange multipliers. The saddle point can be found by setting the derivatives equal to zero. After simplifying, we obtain the following optimization problem:

$$\max \left(-\frac{1}{2} \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^T \sum_{j=1}^T \alpha_{ki} \alpha_{lj} y_i y_j K(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(l)}) + \sum_{k=1}^n \sum_{i=1}^T \alpha_{ki} \right) \quad (39)$$

subject to

$$\sum_{k=1}^n \sum_{i=1}^T \alpha_{ki} y_k = 0, \quad (40)$$

$$\sum_{i=1}^T (\beta_{ki} + \alpha_{ki}) = C h_k(t), k = 1, \dots, n, \quad (41)$$

$$C F_i \leq \sum_{k=1}^n (\beta_{ki} + \alpha_{ki}) \leq C \sum_{k=1}^n G_i. \quad (42)$$

Let us introduce new variables $\gamma_{ik} = (\beta_{ki} + \alpha_{ki}) / C$. Then the above constraints except for the first one are rewritten as

$$F_i \leq \sum_{k=1}^n \gamma_{ki} \leq G_i, \sum_{i=1}^T \gamma_{ik} = h_k(t), \quad (43)$$

$$\alpha_{ki} \leq C \gamma_{ki}, i = 1, \dots, T, k = 1, \dots, n. \quad (44)$$

In sum, we have derived a new optimization problem for computing the optimal values of α_{ki} and γ_{ik} . It is very interesting to see that constraints for γ_{ki} repeat the constraints for π_i . This is an important property of the obtained optimization problem. The optimal solution at the t -th iteration will be denoted as $\alpha_{ki}^{(t)}$.

The separating function at the t -th iteration is of the form:

$$f^{(t)}(\mathbf{x}) = \sum_{k=1}^n \sum_{i=1}^T \alpha_{ki}^{(t)} y_k K(\mathbf{x}_i^{(k)}, \mathbf{x}) + b^{(t)}. \quad (45)$$

After substituting the optimization problem (39)-(44) into the AdaBoost (Step 3 of Algorithm 1), we get the double adaptation. The first one is the updating of weights of observations. The second adaptation is the change of the sets of weights $\mathcal{P}_k^{(t)}$.

6 Conclusions

Three adaptive minimax SVM-based algorithms have been proposed in the paper. The first one can be regarded as a special case of the second algorithm. The second one can be also regarded as a special case of the third algorithm. The main peculiarity of the algorithms is that they use sets of weights instead of their precise values which are usually applied in many classification algorithm. The sets of weight are caused by the introduced transformation of uncertain set-valued training data from many robots to training data with the weight uncertainty in the form of these sets. The second peculiarity of one of the algorithms is that it is based on the use of Walley's imprecise Dirichlet model which allows us to reduce the sets of weights by repeating the classification procedure many times. An important property of the IDM is that it takes into account the prior total ignorance about weights before getting observations. The third peculiarity of the algorithms is their adaptivity. The sets of weights assigned to robots are adopted to classifiers. The fourth peculiarity is that the algorithms are robust because they use the minimax strategy for dealing with the weighted empirical risk measure under sets of weights.

It should be noted that the quadratic optimization problems which have to be solved for constructing the proposed classifiers are similar to the standard SVM optimization problems. Their difference is in additional linear constraints that, in fact, restrict sets of weights. In spite of the optimization problem simplicity, the standard software developed for the SVM in many packages, unfortunately, cannot be used. Therefore, the corresponding software has to be developed for implementing the proposed algorithms.

Acknowledgement

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

References

A. Cowley, H.-C. Hsu, and C.J. Taylor. Distributed sensor databases for multi-robot teams. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA '04*, volume 1, pages 691–696, April 2004.

- G.A. Di Caro, A. Giusti, J. Nagi, and L.M. Gambardella. A simple and efficient approach for cooperative incremental learning in robot swarms. In *Proceedings of the 16th International Conference on Advanced Robotics (ICAR-2103)*, pages 1–8, Nov 2013.
- P. Du, J. Xia, W. Zhang, K. Tan, Y. Liu, and S. Liu. Multiple classifier system for remote sensing image classification: A review. *Sensors*, 12(4):4764–4792, 2012.
- Y. Freund and R.E. Schapire. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- A. Khamis, A. Hussein, and A. Elmogy. Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks*, volume 604, pages 31–51. Springer International Publishing, Cham, 2015.
- I. Navarro and F. Matia. An introduction to swarm robotics. *ISRN Robotics*, 2013(Article ID 608164):1–10, 2013.
- A. Pronobis, O.M. Mozos, and B. Caputo. SVM-based discriminative accumulation scheme for place recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2008*, pages 522–529, May 2008.
- A. Ravet, S. Lacroix, G. Hattenberger, and B. Vandepor-taele. Learning to combine multi-sensor information for context dependent state estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5221–5226, Tokyo, Nov 2013. IEEE.
- B. Scholkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Cambridge, Massachusetts, 2002.
- Y. Tan and Z.-Y. Zheng. Research advance in swarm robotics. *Defence Technology*, 9:18–39, 2013.
- L.V. Utkin. The imprecise dirichlet model as a basis for a new boosting classification algorithm. *Neurocomputing*, 151(3):1374–1383, 2015. doi:10.1016/j.neucom.2014.10.053.
- P. Walley. Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B*, 58:3–57, 1996. with discussion.
- S.E. Yuksel, J.N. Wilson, and P.D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012.