Information Sciences 264 (2014) 182–195

Contents lists available at ScienceDirect

Information Sciences

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

Clustering-based ensembles for one-class classification

Bartosz Krawczyk [a,*], Michał Woźniak [a], Bogusław Cyganek [b]

[a] Department of Systems and Computer Networks, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
[b] AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland

ARTICLE INFO

Article history:
Received 8 October 2012
Received in revised form 5 November 2013
Accepted 14 December 2013
Available online 31 December 2013

Keywords:
One-class classification
Multiple classifier system
Classifier ensemble
Clustering
Soft computing

ABSTRACT

This paper presents a novel multi-class classifier based on weighted one-class support vector machines (OCSVM) operating in the clustered feature space. We show that splitting the target class into atomic subsets and using these as input for one-class classifiers leads to an efficient and stable recognition algorithm. The proposed system extends our previous works on combining OCSVM classifiers to solve both one-class and multi-class classification tasks. The main contribution of this work is the novel architecture for class decomposition and combination of classifier outputs. Based on the results of a large number of computational experiments we show that the proposed method outperforms both the OCSVM for a single class, as well as the multi-class SVM for multi-class classification problems. Other advantages are the highly parallel structure of the proposed solution, which facilitates parallel training and execution stages, and the relatively small number of control parameters.

1. Introduction

Well-known and reliable classifiers tend to fail when faced with new problems such as an atypical class distribution, non-stationary environments, or massive data. Therefore, new methods must be developed to deal with the challenges arising and improve the quality of real-life decision support systems.

One of these newly introduced methodologies is known as one-class classification (OCC) [31], which assumes that during the training stage only objects originating from a single class are available. These are called the target concept and are denoted by ω_T. The purpose of OCC is to calculate a decision boundary that encloses all available data samples, thereby describing the concept [53]. During the execution phase, new objects, unseen during training, may appear. These may originate from one or more distributions and represent data outside the target concept. Such objects, denoted by ω_O, are referred to as outliers.

For a single OCC classifier it may be difficult or even impossible to find a good model owing to limited training data, high feature space dimensionality, and/or the properties of the particular classifier. To avoid a too complex model and overfitting of the training target data, a simpler model with a lower number of features or one that has been trained with smaller chunks of data, can be created. Although the complexity of such a model is reduced, the quality thereof also declines significantly. However, it has been shown that a group of individual OCC models can help alleviate the aforementioned problems.

Here one may use an approach known as multiple classifier systems (MCSs), which is considered to be one of the fastest growing fields in machine learning [26]. MCSs are based on the idea of combining several classifiers into a compound recognition system that can exploit the strengths of individual predictors [60]. Each classifier may output a different decision boundary, and so have different competence areas over the analyzed dataset [7]. When combined, the collective decision

* Corresponding author. Tel.: +48 71 320 35 39; fax: +48 71 320 29 02.
E-mail addresses: bartosz.krawczyk@pwr.wroc.pl (B. Krawczyk), michal.wozniak@pwr.wroc.pl (M. Woźniak), cyganek@agh.edu.pl (B. Cyganek).

0020-0255/$ - see front matter
http://dx.doi.org/10.1016/j.ins.2013.12.019

accuracy can outperform any of the individual predictors. However, several important issues, such as selecting the individual classifiers, as well as choosing a fusion method to establish a group decision, must be considered when designing an MCS. Classifiers used to create the ensemble in an ideal situation should be highly accurate and complement each other (i.e., the ensemble should display high diversity). Adding classifiers that are not diverse with respect to those already in the pool will not improve the accuracy of the compound classifier, but will only increase the overall computational cost [5]. It is worth noting that combination rules, for example, majority voting, could even lead to a deterioration in performance of the ensemble of classifiers [36]. On the other hand, building an MCS with highly diverse but poor quality classifiers will result in a weak committee. Therefore, classifier selection is a critical step in the ensemble design process [15].

MCSs are an attractive yet still largely unexplored, alternative for OCC problems. Most of the works concentrate on practical applications of OCC ensembles. Much still needs to be done to gain insight into the theoretical background to this problem, as well as to draw conclusions on how to build efficient OCC ensembles regardless of the intended application [35].

We propose an approach based on the idea of data clustering in the feature space. OCC models are built based on each of the clusters. In this way we ensure that the pool of predictors is highly diverse and mutually complementary (owing to training on different inputs, i.e., clusters of training objects). This can be seen as an extension of the popular family of ensembles derived from the idea of *clustering and selection* proposed by Kuncheva [37]. So far, two other research teams have worked on this topic, proposing very simple hybrid methods for combining clustering and OCC [38,45].

The contributions of this work are as follows:

- We propose building an ensemble of one-class classifiers based on clustering of the target class. This ensures initial diversity among the classifiers in the pool (as they are based on different inputs) and the correct handling of possible issues embedded in the nature of data, such as a rare distribution or chunks of objects.
- We propose an elastic and efficient framework for this task, which requires only the selection of several components, namely, the clustering algorithm, individual classifier model, and fusion method. These can easily be chosen by the user, as there are practically no limitations on their nature. All other parameters for the method are selected automatically.
- We discuss the possibility of extending our one-class ensemble to an efficient tool for multi-class problems.
- We carry out extensive computational tests on a diverse set of benchmarks that highlight the influence of component selection on the overall method quality and show that the proposed approach outperforms the standard OCC methods as well as a single multi-class support vector machine (SVM) in multi-class classification problems.

Our ensemble is easy to use in many practical applications where it is difficult or even impossible to obtain counter-examples (e.g., machine fault diagnosis), or where, owing to a complex data distribution, the class decomposition approach can lead to a significant improvement in recognition quality over the well known multi-class approaches (e.g., imbalanced classification).

This paper is organized as follows. In the next section the idea of OCC is presented. In Section 3 the architecture of the proposed compound recognition system is explained. The components that must be selected as input for the system are also presented. In Section 4 the experimental results are presented and discussed. The paper ends with the presentation of our conclusions in Section 5.

2. One-class classification

OCC aims to distinguish the target concept objects from possible outliers, and hence it is often referred to as learning in the absence of counter-examples. Although OCC is quite similar to binary classification, the primary difference lies in how the one-class classifier is trained. In standard dichotomy problems it is expected that objects from the other classes tend to come from one direction. Here the available class must be separated from all the possible outliers, which leads to a situation in which a decision boundary must be estimated in all directions in the feature space around the target class. An example OCC problem is depicted in Fig. 1.

OCC is a solution to many real-life problems where there are abundant data for a single class, but it is difficult or even impossible to obtain data for other objects. This is often the case in problems such as image analysis [49], intrusion detection [23], machine fault diagnosis [2], and solid-state fermentation [27].

It is worth noting that there are two different views of OCC:

- As a tool for solving single class problems, where during training only data drawn from the target concept are available e.g., for web page classification [63].
- As a method for the decomposition of multi-class decision task into simpler ones. A canonical decomposition is conducted with the use of binary classifiers [20]. In this approach an M class problem is solved by combining M one-class classifiers, each of which is responsible for the recognition of a different class.

Several methods dedicated to solving OCC problems have recently been introduced. Based on the literature, three main approaches can be distinguished:

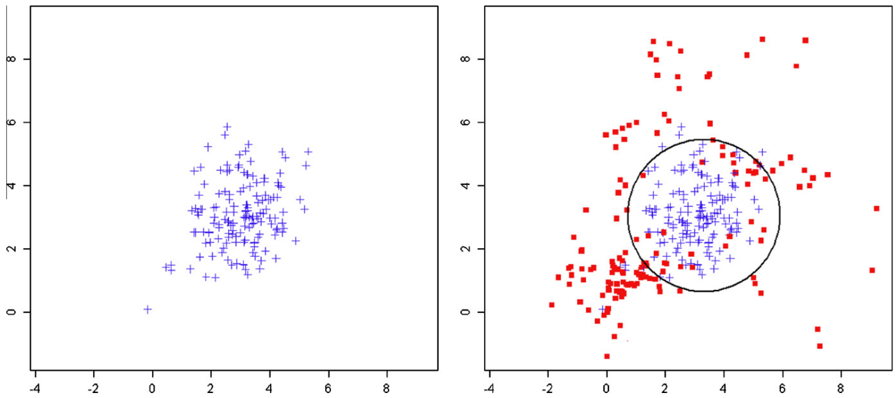

Fig. 1. The concept of one-class classification. (*Left*) Data available during classifier training (blue) representing the target concept. (*Right*) Boundary one-class classifier enclosing all the relevant samples and outlier objects (red) that appeared during the execution of the model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- The first approach, which is simple and can be effective in some cases, comprises methods based on density estimation of a target class. However, this approach has limited application, as it requires a high number of available samples and the assumption of a flexible density model [47]. The Gaussian model, the mixture of Gaussians [65], and the Parzen density [12] are among the most popular density methods for OCC.
- Reconstruction methods, originally introduced as a tool for data modeling [11], constitute another group of algorithms, which make assumptions about the object distribution. Use of reconstruction methods for OCC is based on the idea that it is possible that unknown outliers do not satisfy these assumptions about the structure of objects under consideration. The most popular techniques are the *k*-means algorithm [9], self-organizing maps [54], and auto-encoder networks [40].
- Estimating the complete density or structure of a target concept in a one-class problem can very often be too demanding or even impossible. Therefore, boundary methods have been proposed in recent years. These concentrate on estimating only the close boundary for the given data, assuming that such a boundary will sufficiently describe the target class [29]. The main aim of these methods is to find the optimal size of the volume enclosing the given training points [52], because a too small volume can lead to an overtrained model, whereas one that is too large may lead to extensive acceptance of outliers into the target class. Since these methods rely heavily on the distance between objects, proper feature scaling is a very important data preprocessing step. On the other hand, boundary methods require a smaller number of objects to estimate the decision criterion correctly compared with the two previous groups of methods. The most popular methods in this group include the support vector data description [50] and the one-class support vector machine [10]. The presented work focuses on this last method.

There are several papers dealing with a combination of one-class classifiers [42,58] and their practical application [23]. Ensembles are a promising research direction for OCC problems [35], as they allow us to train less complex individual classifiers, thereby reducing the risk of model overfitting, which is one of the major concerns in using OCC. Additionally, they are an ideal solution for implementation in a distributed environment; most of the OCC classifiers (especially the boundary-based ones) are computationally expensive, and therefore, relying on several weak models that run independently may significantly reduce the training cost of the recognition system. It has been shown that MCS for OCC, designed on the basis of the random subspace method [24,21], can outperform a single-model approach [33] and can introduce novel diversity measures dedicated to pruning the OCC ensembles [32]. These approaches assume dimensionality reduction of the base classifiers.

The one-class boundary methods are based on computing a distance between object x and the boundary enclosing the target class ω_T. This allows the use of fusion methods based on discrete outputs (returned class label) of the individual classifiers – as is the case in the voting methods [56]. However, to apply more sophisticated fusion methods, which assume continuous outputs for each of the individuals, the support of an object x for a given class is required.

We propose using the following heuristic support function based on the distance:

$$F(x,\omega_T) = \frac{1}{c_1} exp(-d(x|\omega_T)/c_2), \tag{1}$$

which models a Gaussian distribution around the classifier, where $d(x|\omega_T)$ is the distance (in this study Euclidean distance was used) from the evaluated object to the support vectors describing the target concept, c_1 is a normalization constant, and c_2 is the scaling parameter. Parameters c_1 and c_2 should be fitted to the target class distribution.

It is worth pointing out that in our other work [34] we have shown that the Euclidean distance is not always the best choice for creating support functions for OCC ensembles. However, so far no systematic trend has been found. We do not address this problem further in this paper.

3. Details of the proposed method

3.1. Architectures for the proposed method

In this paper we propose a new architecture for creating ensembles of one-class classifiers based on the clustering of a feature space into smaller partitions. Additionally, we incorporate our new compound classifier into an architecture that allows both one-class and multi-class problems to be solved. Therefore, in this section we describe our algorithm from two different perspectives – a local perspective (the details of the introduced one-class clustering based ensemble) and a global perspective (the overall architecture for problem decomposition).

First we concentrate on the latter architecture, which we call the one-class clustering-based ensemble (OCClustE). The idea behind OCClustE stems from our previous works where we showed that increasing the number of one-class classifiers assigned to the target class can often lead to a significant improvement in classification accuracy [33]. The main problem with such an approach is how to create a pool of accurate and diverse one-class classifiers. Previously, for this task, we applied the random subspace method, which creates several classifiers based on the subspaces of randomly selected features. In another of our works, we showed that data partitioning by clustering can be an effective preprocessing step for further training of one-class classifiers in an image segmentation task [13].

In this paper we propose using a clustering algorithm to partition the feature space. In the next step each of these clusters is used to train a one-class classifier. This leads to the formation of a pool of K classifiers assigned to the mth class, as follows:

$$\Pi_m = \{\Psi_m^{(1)}, \Psi_m^{(2)}, \ldots, \Psi_m^{(K)}\}. \tag{2}$$

Through this we achieve the goal of creating a pool of several one-class predictors for each of the target classes, and at the same time we ensure their diversity (as a result of using different inputs in their training), which leads to better performance of the ensemble. Subsequently, a classifier fusion method combines the outputs of the classifiers to deliver a final (local) decision. An overview of the proposed OCClustE method is illustrated in Fig. 2.

Differences between the single OCC and OCClustE models for an example dataset are presented in Fig. 3. In summary, the approach proposed in this paper leads to several improvements compared with the standard OCC models:

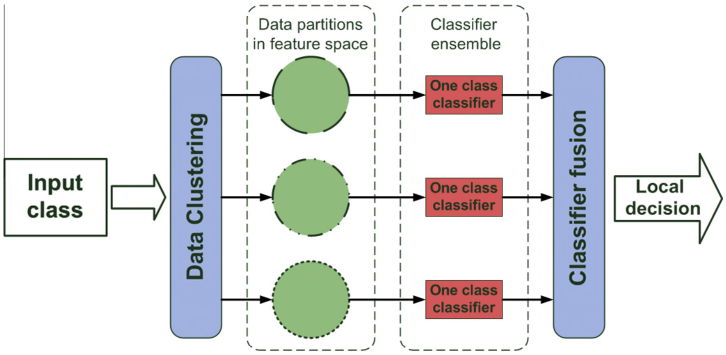

Fig. 2. Overview of the OCClustE.

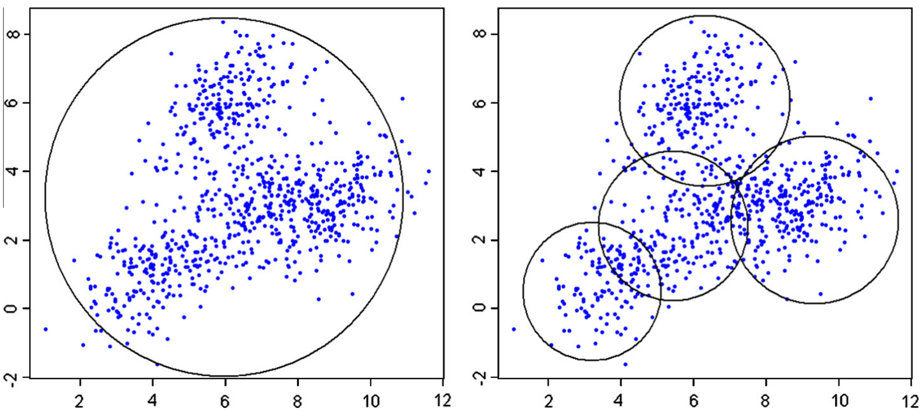

Fig. 3. Differences between the outputs of a standard approach and the proposed one for a one-class toy problem. (*Left*) Target concept enclosed by a single model approach. (*Right*) Target concept after OCClustE classification with four clusters.

- Boundary approaches (such as the one-class support vector machine (OCSVM)) were proven to have better generalization abilities than clustering-based (reconstruction) OCC [50], but are prone to atypical data distributions. Therefore, a hybrid method utilizing both approaches combines the advantages of each while reducing their drawbacks.
- As each classifier is trained only on a partition of the data, its complexity is lower than in the case of a single model approach. This leads to reduced probability of overtraining the predictor. Additionally, a number of individual classifiers can easily be applied in a distributed environment [43], leading to a significant decrease in execution time.
- Partitioning ensures the initial diversity and mutual complementariness of the classifier ensemble.
- Using chunks of data as the classifier input leads to a reduction in the problem known as the empty sphere; that is, the area covered by the boundary in which no objects from the training set are located [28].
- A boundary classifier trained on a more compact data partition usually has a lower number of support vectors.

In the case of a single-class problem, only one OCClustE model is created. Yet, this approach can easily be applied to solving multi-class problems. We now present the architecture for a multi-class classification system based on class decomposition with the local OCClustE.

The proposed architecture for multi-class problems comprises three main steps:

1. Class decomposition: in this step an M-class problem is decomposed into M one-class problems. This approach is valid for multi-class classification. In the case of single-class classification, the decomposition can be omitted as we already have a one-class problem as the input.
2. Classification: in this step each of the classes is considered to be an independent recognition task. To solve each of these, the OCClustE algorithm is employed. Therefore, we have M local OCClustE models, each assigned to a different class:

$$\Pi = \{\Pi_1, \Pi_2, \ldots, \Pi_M\}, \tag{3}$$

where Π_m is the OCClustE model assigned to the mth class and Π is the pool of M OCClustE classifiers.

3. Fusion: after the classification step we have M separate local decisions, one for each of the classes in the problem under consideration. Therefore, each of the local ensembles outputs whether the considered object x belongs to its target class ω_T or is an outlier. In this step the original multi-class problem is reconstructed by the fusion method. In the case of single-class problems the output of the local ensemble trained on the target class is also the global output of the whole system.

An outline of the global approach is presented in Fig. 4.

Differences between single-model fusion and the OCClustE global multi-class approach are illustrated in Fig. 5.

Using the proposed architecture for multi-class data decomposition leads to a significantly smaller overlap between the OCC predictors assigned to each of the classes.

This begs the question: why use an ensemble of one-class classifiers in cases where objects from many classes are available at the training stage? The main advantages of using one-class classifiers are seen in problems for which not all classes are known, or those with highly imbalanced data distributions (e.g., there are large differences in the available training patterns for different classes). Additionally, using one-class classifiers can lead to a different decision boundary; multi-class classifiers search for a optimal separation boundary, while one-class methods focus on capturing properties of a given class. Therefore, the former can be used in cases with high class overlap or where the class distribution is spread over several disjoint data chunks. There are many examples of such classification problems, e.g., in computer vision the appearance model of the object being tracked is often known, whereas those of other objects that can be encountered in the images are unknown. In some other cases, gathering training data is possible only for selected conditions of the system, e.g., when collecting data for normal engine operation it is generally not possible to collect data for failure conditions, which are rare and expensive to simulate. In these cases interpretability is also much better since other classes may not even be known. In several situations e.g., chemometrics [48], we may have several classes at our disposal during the training phase. Yet, during the execution

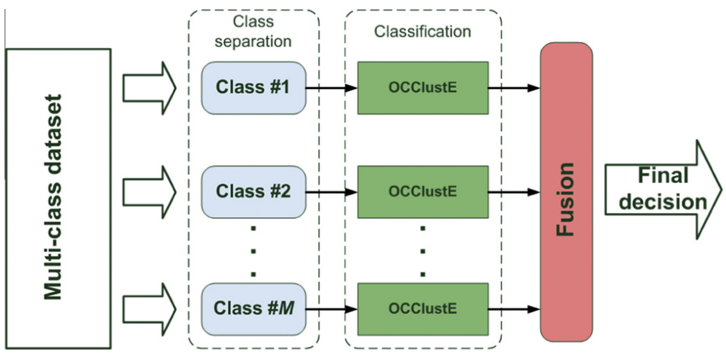

Fig. 4. Overview of the global architecture of the proposed compound classifier model.

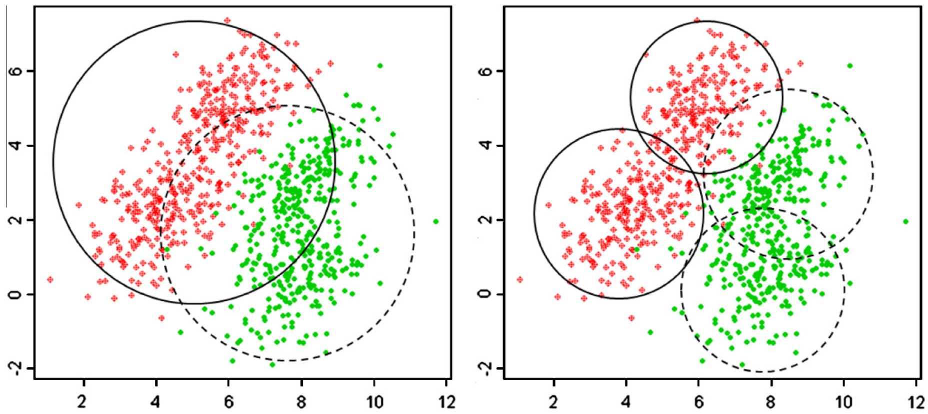

Fig. 5. Differences between the fusion of single models assigned to each of the classes and the OCClustE for a multi-class toy problem. (*Left*) Binary problem solved by fusion of two one-class classifiers. (*Right*) Two-class OCClustE output with two clusters per class.

phase, frequent outliers, concept drift, or new classes may appear. Canonical multi-classification algorithms tend to fail under such conditions, whereas one-class ensembles are robust to such situations. Therefore, our proposed method combines the high recognition rate for multi-class problems with improved robustness to unexpected changes in data.

It should, however, be noted that dependencies among classes have a high impact on the performance of each ensemble. One-class classifiers focus on capturing the properties of the target class, and therefore can deal with difficulties embedded in the nature of the data and, to some extent, with dependencies among classes. In the case of high overlap, the performance of the ensemble deteriorates. Therefore, we propose embedding certain methods to deal with such issues. Using kernel functions in both the clustering (kernel fuzzy *c*-means) and classification (kernel in a weighted OCSVM (WOCSVM)) steps can alleviate some unwanted dependencies between the classes. Additionally, by using the WOCSVM, we can control the degree of influence of difficult objects on the shape of the decision boundary, thereby reducing the potential overlap regions among the classifiers.

3.2. Components of the OCClustE

As discussed in the previous section, the proposed framework is very flexible, as it places no restrictions on the nature of its three main components:

- Clustering: any clustering algorithm can be applied to the OCClustE. However, the algorithm should be chosen carefully as data partitioning has a critical impact on the classification step, since badly defined clustering objectives may lead to the formation of a pool of weak classifiers with low diversity. Additionally, the correct number of clusters should be selected for the considered problem. A proposal on how to achieve this is presented in a later section.
- Classification: the OCClustE is designed to work with one-class classifiers, especially those based on boundary estimation. However, the choice of the type of classifier is left to the end-user of the proposed method.
- Fusion: classifier fusion methods must be chosen for two purposes: first, to combine the outputs of individual classifiers in the OCClustE ensemble, and second (in the case of a multi-class decomposition) to combine the local outputs of each of the OCClustE ensembles to reconstruct the original multi-class task from several one-class ones.

Next, we discuss a possible setting of the aforementioned components of the OCClustE, which was used in the experiments to assess the quality of the proposed method.

3.2.1. Clustering algorithms

In this study we investigated the behavior of three clustering algorithms:

- *k*-means [39],
- fuzzy *c*-means [4,57], and
- kernel fuzzy *c*-means, which is a modification of the fuzzy *c*-means algorithm that operates in an artificial feature space created by a kernel function [64].

Additionally, we propose a simple and effective measure for assessing the number of clusters.

The method for assessing the quality of segmentation is very important, especially if there is no prior information on the possible number of clusters. Usually, the ratio between cluster compactness and cluster separability is evaluated. On the other hand, the entropy of the membership values, which depends on the data and the number of clusters C, is a good indicator of the quality of clustering [3]. This is computed as:

$$E(C) = -\sum_{c=1}^{C}\sum_{i=1}^{N} w_{ci} \log w_{ci}, \tag{4}$$

where N denotes the number of data points and w_{ci} is the weight assigned to a given cluster. If the number of clusters C is not known in advance, clustering can be performed for a varying numbers of clusters, and the number with minimal entropy can be chosen to build the ensemble [14]. Such a strategy usually provides a useful indication of the number of means, which is data dependent. We acknowledge that the entropy criterion is not a perfect solution for estimating the number of clusters. Yet, at the same time, it is less time-consuming than manually checking the correlation between the number of clusters and final accuracy. As our aim is to create an ensemble classifier that will require minimal intervention with regard to parameter settings from the end-user, the entropy criterion seems an attractive solution.

As we are dealing with multi-class problems, the entropy created by the number of clusters may be different for each of the classes. We propose simplifying this problem by assuming that the number of clusters is the same for each of the classes, and therefore we use the mean of the entropy computed over all classes as follows:

$$\widehat{E}(C) = \frac{1}{M}\sum_{m=1}^{M} E(C_m), \tag{5}$$

where $\widehat{E}(C)$ is the mean value of the entropy and $E(C_m)$ is the entropy value for the mth class calculated according to (4).

3.2.2. Classification algorithms

In this study we used two one-class classification models, based on the estimation of the decision boundary: a popular OCSVM and a modification thereof that assigns weights to each of the training points:

- One-class support vector machine
  The OCSVM [44] can deal with datasets that contain patterns from only a single target class. OCSVM classification aims to discriminate one class of target samples from all the others, which entails learning the minimum volume contour that encloses most of the data in a given dataset. Its original application was in outlier detection, that is, finding data that differ from most of the examples within a dataset. A graphic portrayal of the OCSVM method is presented in Fig. 6.
  OCSVM uses the training data to learn a function $f_\chi : \mathbb{R}^d \mapsto \mathbb{R}$ such that most of the data in χ belong to the set $\mathcal{R}_\chi = \{x \in \mathbb{R}^d; f_\chi(x) \geqslant 0\}$ and the volume of $\mathcal{R}_\chi$ is minimal. This problem is known as *minimal volume set* (MVS) estimation. Because we are considering an M-class recognition problem, we have to learn M membership functions f_{χ_i} – one for each class.
  OCSVM uses the kernel function $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ to estimate the MVS. In our research we use a Gaussian radial basis function (RBF) kernel. OCSVM has been successfully applied to many practical problems, such as image retrieval [10] and brain functionality analysis [22].
- Weighted one-class support vector machine
  To allow some outliers in the training set and to make the classifier more robust, we allow the distances between some training points and the center of the hypersphere to be greater than r, incurring some additional penalty factor. For this purpose slack variables ξ_i are introduced. In the work by Bicego et al. [6] this concept was further extended by introducing weights w_i, thereby allowing an importance measure to be associated with each of the training data. The idea was to make the penalty factor, expressed by the slack variables ξ_i, to be additionally controlled by w_i. Hence, if x_i is associated with a small weight w_i, the corresponding slack variable ξ_i indicates a small penalty. In effect, the corresponding slack variable ξ_i will be greater, allowing x_i to be located further from the center of the hypersphere. This diminishes the impact

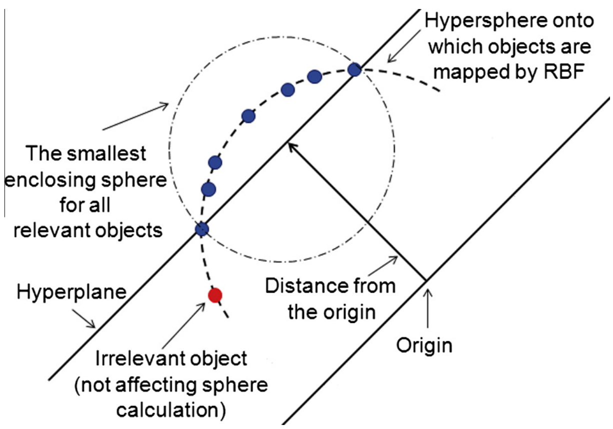

Fig. 6. The OCSVM classifier, in which relevant training points are mapped onto the smallest enclosing hypersphere.

of x_i on the center and the radius of the hypersphere.

In our previous work [14] we proposed taking the weight values from the membership functions given by the fuzzy c-means and kernel fuzzy c-means algorithms. The weights should be normalized; i.e., for a given data point the sum of all its weights across the clusters should be one. However, for a chosen cluster its sum across all data points can take any value in the range (0,N), and therefore the points with w_i close to zero are excluded from the training as being unimportant for that cluster (in practice we set a threshold of $1e-09$). After the set of weights has been computed in the clustering stage, parameter B, which controls the optimization process, is chosen. That is, for each cluster m it needs to be greater than the inverse of the sum of all the weights w_{mi} [14].

3.2.3. Fusion algorithms

In this study we considered three fusion approaches for one-class classifiers. The following methods were investigated, assuming L classifiers in a pool.

- **Majority voting** One of the most popular classifier fusion methods is the majority voting scheme [61], which assigns the label of the class predicted by the highest number of classifiers from the pool to a new object x.
- **Maximum support** Tax and Duin [51] proposed five different methods using fusion of support functions for combining one-class classifiers. We extended their proposal by using a simple max rule, which selects the class with the highest support given by the OCC from the pool.
- **Error-correcting output codes** Error-correcting output codes (ECOC) [16] is a simple yet effective framework for dealing with multi-class problem reconstruction from the decisions of binary classifiers. The basis of the ECOC framework consists of designing a codeword for each of the classes. These codewords encode the membership information of each class. Arranging the codewords as rows in a matrix, we obtain an encoding matrix. Each of these binary problems (or dichotomizers) splits the set of classes into two partitions (coded as +1 or -1 according to their class set membership or 0 if the class is not considered in the current binary problem). Then, in the decoding step, by applying the trained classifiers, a code is obtained for each data point in the test set. This code is compared to the base codewords of each class in the encoding matrix and the data point is assigned to the class with the closest matching codeword [59]. ECOC can easily be used in an OCC ensemble, as we can map the target class as +1 and the unknown outlier class as -1 [58].

3.3. Computational complexity

Computational complexity of the proposed ensemble of classifiers is analyzed differently for the training and testing stages. The training stage is divided into the data clustering and ensemble training parts. For a fixed number of assumed clusters C and dimension of data d, the computational complexity of the family of k-means methods is polynomial with respect to the number of data N [25]. On the other hand, complexity of the sequential minimal optimization (SMO) algorithm used to train the WOCSVM classifiers is $O(LN)$ where L denotes the average number of support vectors in use during the iterations [18].

However, as previously mentioned, the proposed system is versatile and allows the clustering method to be easily changed to better suit the type of input data. Thus, a modified k-means method can be applied, as proposed by Pelleg and Moore [41] or Frahling and Sohler [19], to name but a few. Alternatively, for very large amounts of data or high dimensional data, another clustering method can be used. This is left for future research.

On the other hand, in the testing stage, for a fixed number of support vectors, the computational complexity of the ensemble is linear and depends on the computational complexity of the kernel functions used in the WOCSVM classifiers. Thus, for many practical problems, such as image segmentation, the method allows real-time processing as shown in one of our previous papers [14].

However, as alluded to previously, both the training and testing stages of the proposed method can easily be parallelized. This property can be used to further speed up execution of the training and testing stages of the method.

4. Experimental investigation

In this section, we present the results of thorough experimental investigation examining the behavior of the proposed one-class ensemble approach. The aim of the experiments was to assess the quality of the OCClustE components tested (clustering methods, classification algorithms, and fusers) and to compare the proposed method with known approaches for multi-class decomposition using one-class classifiers, i.e., where a single one-class classifier is assigned to each of the classes.

Our aim is to evaluate whether further partitioning of the target class will lead to an improvement in recognition accuracy and to ascertain how well our method works with multi-class datasets.

4.1. Experimental setup

All experiments were carried out in the R environment [55]. Computer implementations of the clustering and classification methods used were taken from dedicated packages (if available) built into the above mentioned software. This ensured that the results achieved the best possible efficiency and that performance was not diminished by implementation issues. Other methods not provided by dedicated packages were implemented by the authors.

We use the native R implementation of the k-means algorithm, while the fuzzy c-means and kernel fuzzy c-means implementations were taken from the e1071 package [17] and *kernlab* package [30], respectively.

Two models of one-class classifier were used: the OCSVM with RBF kernel, taken from [8], and the modified weighted version thereof, implemented by the authors. WOCSVM was used only with the fuzzy clustering methods; similar to our proposed method it uses membership values to calculate object weights. On the other hand, for a crisp k-means the methods are equivalent.

We compared our approach with the standard OCC method (that is, assigning a single OCC classifier to each of the classes) and with the multi-class SVM with RBF kernel, trained using the SMO procedure and utilizing the one-versus-one scheme for multi-class datasets. Additionally, for comparison we used the boosted one-class ensemble, dedicated to multi-class classification [62], which used an identical base model as our approach and a pool consisting of 20 classifiers.

The combined 5×2 cv F test [1] was carried out to assess the statistical significance of the obtained results.

4.2. Datasets

In the experiments, we concentrated on the usage of OCClustE for decomposition of multi-class problems. In total we selected 20 multi-class datasets, 19 of which were from the UCI Repository with the final one from the chemoinformatics domain, describing the process of discovering pharmaceutically useful isoforms of the CYP 2C19 molecule. The latter dataset is available for download at [46]. Details of the chosen datasets are given in Table 1.

4.3. Selecting the number of clusters

The entropy criterion, described in Section 3.2.1, was used to establish the number of clusters for each of the 20 datasets. We calculated the entropy criterion for each of the three examined clustering methods, with the results presented in Figs. 7–9, respectively.

4.4. Results and discussion

Results of the computational experiments are presented in Table 2. A multi-class SVM, a standard OCC approach (single one-class classifier assigned to each of the classes), and boosted OCC were used as reference methods.

Analysis of the results clearly shows that the proposed method displays high quality. In 16 of the 20 benchmark tests the OCClustE significantly outperformed the single OCC predictors. Additionally, in 13 cases it performed better than the multi-class SVM. This is a very interesting result as the SVM had all the data available during the training process, whereas the OCClustE relied on independent recognition of a simplified problem. This confirms our previous statement that OCC ensembles may be a useful tool for multi-class decomposition [58], especially in the case of imbalanced datasets.

Table 1
Details of datasets used in the experiments. (Numbers in parentheses denote the number of objects in the minority class in the case of binary problems.)

No.	Name	Objects	Features	Classes
1	Audiology	226	69	24
2	Balance	625	4	3
3	Breast-cancer	286 (85)	9	2
4	Breast-Wisconsin	699 (241)	9	2
5	Colic	368 (191)	22	2
6	Credit-rating	690	15	6
7	Diabetes	768 (268)	8	2
8	Glass	214	9	6
9	Heart-c	303	13	5
10	Heart-h	294	13	5
11	Heart-statlog	270 (120)	13	2
12	Hepatitis	155 (32)	19	2
13	Ionosphere	351 (124)	34	2
14	Iris	150	4	3
15	Lymphography	148	18	4
16	Primary tumor	339	17	21
17	Sonar	208 (97)	60	2
18	Voting records	435 (168)	16	2
19	Wine	178	13	3
20	CYP2C19 isoform	837 (181)	242	2

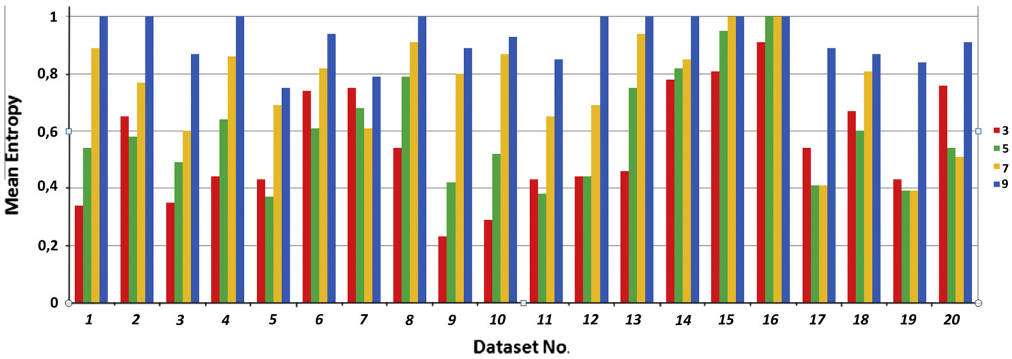

Fig. 7. Mean entropy values over 20 datasets for *k*-means with the number of clusters {3; 5; 7; 9}.

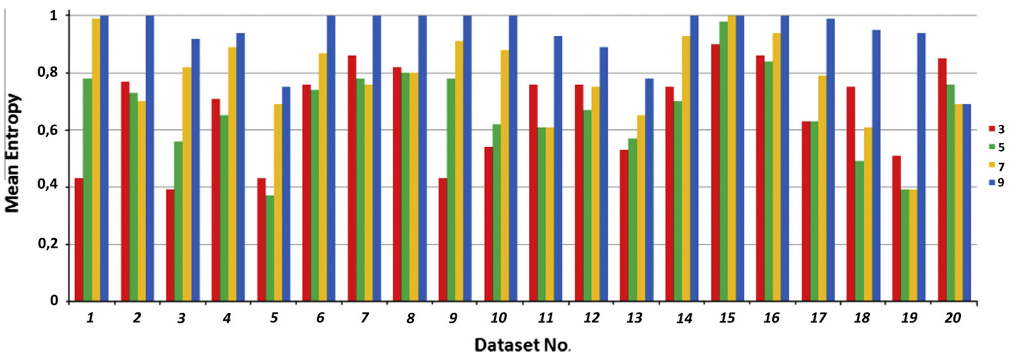

Fig. 8. Mean entropy values over 20 datasets for fuzzy *c*-means with the number of clusters {3; 5; 7; 9}.

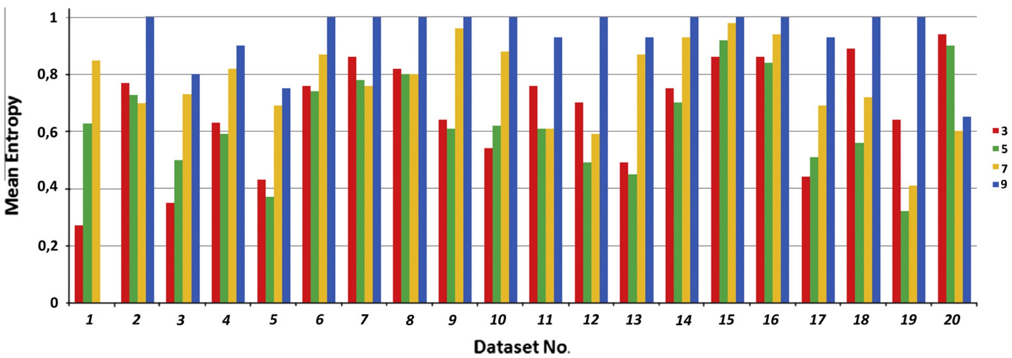

Fig. 9. Mean entropy values over 20 datasets for kernel fuzzy *c*-means with the number of clusters {3; 5; 7; 9}.

WOCSVM outperformed the standard crisp model in 17 of the 20 cases. This shows that inclusion of weights based on the clustering membership function can lead to more diverse, and at the same time highly accurate, predictors. As expected, kernel fuzzy clustering outperformed the other simpler methods.

Use of the entropy criterion allowed for the automatic determination of the most promising number of clusters for an ensemble. The experimental results confirm that this criterion coped well with the proposed classification architecture, eliminating the time-consuming manual tuning phase. Of course, the entropy criterion is merely a suggestion for the number of clusters and better results may be achieved after manual experimentation with the settings; this was, however, not our goal. We aimed to create an ensemble classifier that would be easy for the end-user to use. To this end, automatic cluster selection by means of the entropy criterion worked satisfactorily. In future work, we would like to explore the possibility of using clustering methods that do not require a priori knowledge about the number of groups in the data, e.g., DBSCAN.

We tested three fusion methods, each of which represented different groups of combined approaches. We examined the behavior of fusers based on discrete (majority voting) and continuous (max and ECOC) outputs of the base classifiers. The

Table 2
Performance of the proposed method compared with reference algorithms over 20 datasets. Small numbers below the accuracy of a method denote the indexes of those reference methods, the accuracy of which is statistically significantly worse than that presented. 1*pC* represents the approach in which a single one-class classifier is assigned to each of the classes, *OCBoost* denotes boosting one-class support vector machines for multi-class classification, *km* for *k*-means, *fcm* for fuzzy *c*-means, *kfcm* for kernel fuzzy *c*-means, *MV* for majority voting fusion, *Max* for maximum support fusion, and *ECOC* for error-correcting output codes fusion.

No.	SVM[1]	Fusion	1pC[2]	OCSVM			WOCSVM		OCBoost[8]
				km[3]	fcm[4]	kfcm[5]	fcm[6]	kfcm[7]	
1.	80.77	MV	78.87	77.23	67.23	77.23	80.24	82.05	86.32
				-	-	-	1,2,3,4,5	1,2,3,4,5	
		Max	83.23	85.02	84.76	85.49	86.32	87.85	
				1	1	1	1,2	1,2,3,4,5,8	
		ECOC	85.02	87.17	87.17	87.17	87.96	88.67	
				1	1	1	1,2	1,2,3,4,5,8	
2.	80.77	MV	75.34	75.23	77.01	77.25	77.87	78.07	84.03
				-	2	2,3	2,3	2,3	
		Max	81.45	81.94	81.94	81.94	84.21	85.20	
				1	1	1	1,2,3,4,5,8	1,2,3,4,5,8	
		ECOC	82.33	84.03	85.23	85.23	87.11	87.11	
				1,2	1,2	1,2	1,2,3,4,5,8	1,2,3,4,5,8	
3.	69.52	MV	71.23	67.56	68.03	68.43	71.45	72.34	73.78
				-	-	-	1,3,4,5	1,2,3,4,5	
		Max	75.03	74.10	74.23	74.23	75.20	75.20	
				1	1	1	1,8	1,8	
		ECOC	74.35	72.89	73.00	73.78	74.95	76.20	
				1	1	1	1,2,3,4,5,8	1,2,3,4,5,8	
4.	96.57	MV	88.96	87.34	87.56	87.34	92.34	92.04	95.22
				-	-	-	-	-	
		Max	93.21	90.56	90.56	90.56	94.19	94.56	
				-	-	-	-	-	
		ECOC	94.86	94.23	94.67	94.23	96.57	97.01	
				-	-	-	8	8	
5.	82.66	MV	73.89	77.34	77.34	77.76	79.11	80.06	82.98
				2	2	2	2,3,4,5	2,3,4,5	
		Max	77.00	80.83	80.83	81.11	82.95	83.75	
				2	2	2	2,3,4,5	2,3,4,5,8	
		ECOC	77.66	80.54	80.54	80.54	82.98	84.91	
				2	2	2	2,3,4,5	2,3,4,5,8	
6.	84.00	MV	78.02	75.21	75.21	75.56	75.93	76.21	80.90
				-	-	-	-	-	
		Max	80.15	78.03	78.55	78.98	79.34	79.34	
				-	-	-	-	-	
		ECOC	83.30	80.21	80.90	80.23	81.42	81.42	
				-	-	-	2,3,4,8	2,3,4,8	
7.	76.80	MV	75.20	76.32	76.11	76.11	76.98	77.11	75.20
				-	-	-	-	-	
		Max	77.11	79.43	79.43	79.98	80.02	80.98	
				1,2,8	1,2,8	1,2,8	1,2,8	1,2,8	
		ECOC	78.40	79.43	79.43	79.98	80.98	81.78	
				1,2,8	1,2,8	1,2,8	1,2,3,4,5,8	1,2,3,4,5,8	
8.	57.36	MV	55.23	57.21	57.96	57.96	59.20	59.20	57.96
				-	-	-	8	8	
		Max	58.23	60.05	60.05	60.05	61.93	61.93	
				1,2,8	1,2,8	1,2,8	1,2,8	1,2,8	
		ECOC	61.18	62.05	62.60	62.50	63.21	63.21	
				1,2,8	1,2,8	1,2,8	1,2,8	1,2,8	
9.	83.86	MV	70.95	65.23	66.02	66.02	68.12	69.75	69.75
				-	-	-	3,4,5	3,4,5,6	
		Max	73.32	73.95	74.00	74.00	76.20	76.20	
				8	8	8	2,3,4,5,8	2,3,4,5,8	
		ECOC	74.25	78.40	78.40	78.40	81.07	84.25	
				2,8	2,8	2,8	2,3,4,5,8	1,2,3,4,5,6,8	
10.	82.74	MV	77.34	80.12	80.12	80.12	79.32	79.32	79.32
				2	2	2	2	2	
		Max	78.02	80.12	80.23	80.23	80.12	80.12	
				2	2	2	2	2	
		ECOC	78.02	80.95	81.24	81.24	79.10	80.12	
				2,6,8	2,6,8	2,6,8	2	2	
11.	76.89	MV	73.65	76.85	76.85	77.00	78.34	78.95	80.08
				2	2	2	2,3,4,5	1,2,3,4,5	
		Max	75.20	78.83	79.03	78.83	80.08	81.83	
				1,2	1,2	1,2	1,2,3,4,5	1,2,3,4,5,6,8	
		ECOC	75.20	78.83	79.03	78.83	80.08	81.83	
				1,2	1,2	1,2	1,2,3,4,5	1,2,3,4,5,6,8	
12.	85.77	MV	83.11	86.94	86.94	86.94	84.50	85.05	89.13
				1,2	1,2	1,2	2	2	
		Max	88.20	87.70	87.70	87.70	85.20	85.20	
				1,6,7	1,6,7	1,6,7	2	2	
		ECOC	90.17	88.56	89.02	89.13	87.65	88.04	
				1,6,7	1,6,7	1,6,7	2	2	
13.	88.07	MV	86.00	88.79	88.79	88.79	92.35	94.11	96.23
				1,2	1,2	1,2	1,2,3,4,5	1,2,3,4,5,6	
		Max	87.23	92.50	92.80	93.03	95.15	97.32	
				1,2	1,2	1,2	1,2,3,4,5	1,2,3,4,5,6,8	
		ECOC	87.23	92.50	92.80	93.03	95.15	97.32	
				1,2	1,2	1,2	1,2,3,4,5	1,2,3,4,5,6,8	
14.	96.27	MV	88.65	89.80	90.00	90.00	92.30	92.30	94.48
				-	-	-	2,3,4,5	2,3,4,5	
		Max	90.31	93.45	93.80	93.80	95.92	95.92	
				-	-	-	2,3,4,5,8	2,3,4,5,8	
		ECOC	92.27	93.95	93.95	94.48	96.27	96.27	
				-	-	-	2,3,4,5,8	2,3,4,5,8	
15.	84.48	MV	83.64	82.95	83.05	83.70	85.64	86.48	87.03
				-	-	-	1,2,3,4,5	1,2,3,4,5	
		Max	84.02	83.87	83.87	85.23	86.98	87.56	
			-	-	2	1,2,3,4,5	1,2,3,4,5		
		ECOC	85.64	85.64	85.64	86.35	88.21	88.97	
				-	-	2	1,2,3,4,5,8	1,2,3,4,5,8	
16.	47.09	MV	50.05	50.05	50.60	50.60	50.60	50.60	51.63
				1	1	1	1	1	
		Max	51.63	50.60	51.00	51.00	51.63	51.90	
				1	1	1	1	1	
		ECOC	52.72	51.00	51.00	51.00	51.90	52.72	
				1	1	1	1	1,8	
17.	76.60	MV	70.20	65.00	66.05	66.05	67.10	68.45	72.60
				-	-	-	3	3	
		Max	72.60	70.95	71.73	71.73	74.18	75.50	
				-	-	-	2,3,4,5,8	2,3,4,5,8	
		ECOC	72.60	74.95	75.73	75.73	76.18	77.50	
				2	2,8	2,8	2,3,4,5,8	1,2,3,4,5,8	
18.	95.77	MV	83.11	85.23	85.23	85.23	88.09	89.60	88.30
				2	2	2	2,3,4,5	2,3,4,5,8	
		Max	88.30	90.65	90.65	90.65	89.11	89.11	
				2,8	2,8	2,8	2	2	
		ECOC	89.57	91.00	91.50	91.94	90.05	91.00	
				2,8	2,6,8	2,6,8	2,8	2,8	
19.	98.76	MV	94.02	89.07	90.10	90.75	92.20	93.87	92.20
				-	-	3	3,4,5	3,4,5,6,8	
		Max	96.92	93.31	93.31	94.02	95.11	95.98	
				,8	-	3,8	3,4,5,8	3,4,5,8	
		ECOC	96.92	93.31	93.31	94.02	95.11	95.98	
				-	-	3,8	3,4,5,8	3,4,5,8	
20.	72.23	MV	68.02	69.90	69.90	69.90	71.50	71.50	68.02
				2	2	2	2,3,4,5,8	2,3,4,5,8	
		Max	72.47	74.32	74.93	74.93	75.80	79.56	
				1,2,8	1,2,8	1,2,8	1,2,8	1,2,3,4,5,6,8	
		ECOC	76.46	79.03	79.41	79.41	81.17	83.01	
				1,2,8	1,2,8	1,2,8	1,2,3,4,5,8	1,2,3,4,5,6,8	

experimental results confirmed our suspicions that voting-based approaches return inferior results; owing to the distance-based nature of boundary one-class classifiers they tend to work better with continuous outputs (mapped from the distance to the support function). As for the remaining two methods, there is a steady trend over the majority of databases in favor of the ECOC combiner. For the proposed model, the max method will work as a winner-takes-all approach, thus choosing a single best classifier from the pool. ECOC took advantage of all the classifiers in the pool, returning the best results. This shows that after the proposed clustering space split we obtain a pool of mutually supplementary classifiers that benefit from group decision making.

Compared with the one-class boosting algorithm, dedicated to one-class classification, our method is statistically superior in most cases. The authors in [62] showed that their one-class boosting algorithm was highly suited to multi-class datasets, often outperforming canonical classifiers. We achieved better results, especially when using an ECOC fuser. This is most likely due to the fact that our classifiers work on smaller data partitions and therefore the ensemble is able to fully explore their local competencies, easily dealing with underlying issues in class distributions.

Finally, let us look at the comparison of our proposed method and the multi-class SVM. SVMs are natural tools for class decomposition, as they operate on binary problems. Hence, a multi-class problem is split into several simpler problems. In our approach we split several classes into one-class problems. Each method thus offers a different view of the decomposition task. Intuitively, the SVM should yield better performance, as during the training step it has at its disposal objects originating from both classes, whereas our approach does not use counter-examples. Surprisingly, in 13 of the 20 cases our method outperformed the SVM. This leads to the conclusion that the distribution of objects in classes is too complex for a binary kernel classifier to handle. While trying to find a good trade-off among the classes it fails to capture the individual features of each of the distributions. One-class classifiers, on the other hand, work only on a single class, and therefore can more easily adjust to their properties. We can conclude that the clustering step further improves the process of capturing the unique characteristics of the target class by reducing the empty regions within the decision boundary and allowing us to deal more effectively with problems such as sparse distribution, data chunks, or rare objects. A worthwhile research direction would be to compare our approach with other decomposition techniques and different algorithms for multi-class SVMs.

5. Conclusion and future work

This paper presented a method for creating a one-class classifier ensemble based on feature space partitioning. We proposed a two-level architecture for the design of such a classification system. The main advantage of the proposed method is that the combined classifiers trained on the basis of clusters allow us to exploit individual classifier strengths. As a result, these usually outperform traditional methods for one-class classifier combinations for multi-class classification problems operating in a one-versus-one strategy. This observation was confirmed by the computational experiments carried out using a wide range of benchmark datasets. Additionally, during the experiments several different components were tested to find the best setting for the method. Nevertheless, we would like to emphasize that the proposed framework is flexible, and can work with different clustering algorithms and one-class classification methods. Our approach leads to a decrease in the overall training time by distributing the computations for each of the clusters, which could even be executed on different processors.

The results are promising and encourage further research on these topics. We anticipate improvements in our proposed approach by ensuring higher diversity [32] of the classifier ensemble for each of the partitions, e.g., using the random subspace approach [24] and/or by selecting the valuable classifier ensemble using diversity measures dedicated to one-class classifiers [35]. It will also be interesting to apply the OCClustE to a real diagnostic problem to ascertain whether imbalanced classification can be satisfied, for instance, in computer security [23].

Acknowledgments

The work was supported by the Polish National Science Centre under Grant No. N519 576638 for the years 2010–2013, as well as by the Polish National Science Centre Grant No. DEC-2011/01/B/ST6/01994.

References

[1] Ethem Alpaydin, Combined 5 × 2 cv f test for comparing supervised classification learning algorithms, Neural Comput. 11 (8) (1999) 1885–1892.
[2] A. Bartkowiak, R. Zimroz, Outliers analysis and one class classification approach for planetary gearbox diagnosis, J. Phys.: Conf. Ser. 305 (1) (2011).
[3] J. Bezdek, Pattern Recognition With Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[4] J.C. Bezdek, R. Ehrlich, W. Full, Fcm: the fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203.
[5] Y. Bi, The impact of diversity on the accuracy of evidential classifier ensembles, Int. J. Approx. Reason. 53 (4) (2012) 584–607.
[6] M. Bicego, M.A.T. Figueiredo, Soft clustering using weighted one-class support vector machines, Pattern Recogn. 42 (1) (2009) 27–32.
[7] P.P.K. Chan, D.S. Yeung, W.W.Y. Ng, C.M. Lin, J.N.K. Liu, Dynamic fusion method using localized generalization error model, Inform. Sci. 217 (2012) 1–20.
[8] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.
[9] B. Chen, A. Feng, S. Chen, B. Li, One-cluster clustering based data description, Jisuanji Xuebao/Chinese J. Comput. 30 (8) (2007) 1325–1332.
[10] Y. Chen, X.S. Zhou, T.S. Huang, One-class svm for learning in image retrieval, IEEE Int. Conf. Image Process. 1 (2001) 34–37.
[11] K.F. Cheung, Fuzzy one-mean algorithm: Formulation, convergence analysis, and applications, J. Intell. Fuzzy Syst. 5 (4) (1997) 323–332.

[12] G. Cohen, H. Sax, A. Geissbuhler, Novelty detection using one-class parzen density estimator: an application to surveillance of nosocomial infections, Stud. Health Technol. Inform. 136 (2008) 21–26.
[13] B. Cyganek, Image segmentation with a hybrid ensemble of one-class support vector machines, LNAI of Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6076, 2010, pp. 254–261.
[14] B. Cyganek, One-class support vector ensembles for image segmentation and classification, J. Math. Imag. Vis. 42 (2–3) (2012) 103–117.
[15] Q. Dai, A competitive ensemble pruning approach based on cross-validation technique, Knowl.-Based Syst. 37 (2013) 394–414.
[16] Thomas G. Dietterich, Ghulum Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Int. Res. 2 (January 1995) 263–286.
[17] Evgenia Dimitriadou, Kurt Hornik, Friedrich Leisch, David Meyer, Andreas Weingessel, r-cran-e1071, 2011. <http://mloss.org/software/view/94/>
[18] J.X. Dong, A. Krzyzak, C.Y. Suen, A practical SMO algorithm, in: Proc. Int. Conf. on Pattern Recognition, vol. 3, 2002.
[19] Gereon Frahling, Christian Sohler, A fast k-means implementation using coresets, Int. J. Comput. Geometry Appl. 18 (6) (2008) 605–625.
[20] M. Galar, A. Fernndez, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes, Pattern Recogn. 44 (8) (2011) 1761–1776.
[21] N. Garcia-Pedrajas, J. Maudes-Raedo, C. Garca-Osorio, J.J. Rodriguez-Diez, Supervised subspace projections for constructing ensembles of classifiers, Inform. Sci. 193 (2012) 1–21.
[22] A.B. Gardner, A.M. Krieger, G. Vachtsevanos, B. Litt, One-class novelty detection for seizure analysis from intracranial eeg, J. Machine Learn. Res. 7 (2006) 1025–1044.
[23] Giorgio Giacinto, Roberto Perdisci, Mauro Del Rio, Fabio Roli, Intrusion detection in computer networks by a modular ensemble of one-class classifiers, Inf. Fusion 9 (January 2008) 69–82.
[24] Tin Kam Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (August) (1998) 832–844.
[25] Mary Inaba, Naoki Katoh, Hiroshi Imai, Applications of weighted voronoi diagrams and randomization to variance-based k-clustering (extended abstract), in: Symposium on Computational Geometry, 1994, pp. 332–339.
[26] A.K. Jain, R.P.W. Duin, Jianchang Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Machine Intell. 22 (1) (2000) 4–37.
[27] H. Jiang, G. Liu, X. Xiao, C. Mei, Y. Ding, S. Yu, Monitoring of solid-state fermentation of wheat straw in a pilot scale using ft-nir spectroscopy and support vector data description, Microchem. J. 102 (2012).
[28] P. Juszczak, Learning to recognise, A study on one-class classification and active learning, PhD thesis, Delft University of Technology, 2006.
[29] P. Juszczak, D.M.J. Tax, E. Pekalska, R.P.W. Duin, Minimum spanning tree based one-class classifier, Neurocomputing 72 (7–9) (2009) 1859–1869.
[30] A. Karatzoglou, K. Hornik, A. Smola, A. Zeileis, kernlab – an s4 package for kernel methods in r, J. Stat. Softw. 11 (2004) 1–20.
[31] M.W. Koch, M.M. Moya, L.D. Hostetler, R.J. Fogler, Cueing, feature discovery, and one-class learning for synthetic aperture radar automatic target recognition, Neural Netw. 8 (7–8) (1995) 1081–1102.
[32] B. Krawczyk, Diversity in ensembles for one-class classification, in: Mykola Pechenizkiy, Marek Wojciechowski (Eds.), New Trends in Databases and Information Systems, Advances in Intelligent Systems and Computing, vol. 185, Springer, Berlin Heidelberg, 2012, pp. 119–129.
[33] B. Krawczyk, M. Woźniak, Combining diverse one-class classifiers, in: Emilio Corchado, Vaclav Snasel, Ajith Abraham, Michal Wozniak, Manuel Grana, Sung-Bae Cho (Eds.), Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, vol. 7209, Springer, Berlin/Heidelberg, 2012, pp. 590–601.
[34] B. Krawczyk, M. Woźniak, Experiments on distance measures for combining one-class classifiers, in: Proceedings of the Federated Conference on Computer Science and Information Systems, 2012, pp. 89–92.
[35] B. Krawczyk, M. Woźniak, Diversity measures for one-class classifier ensembles, Neurocomputing (2013).
[36] L.I. Kuncheva, C.J. Whitaker, C.A. Shipp, R.P.W. Duin, Limits on the majority vote accuracy in classifier fusion, Pattern Anal. Appl. 6 (2003) 22–31.
[37] Ludmila I, Kuncheva, Clustering-and-selection model for classifier combination, in: KES, 2000, pp. 185–188.
[38] N. Lipka, B. Stein, M. Anderka, Cluster-based one-class ensemble for classification problems in information retrieval, in: SIGIR'12 – Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2012, pp. 1041–1042.
[39] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: L.M. Le Cam, J. Neyman (Eds.), Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, University of California Press, 1967, pp. 281–297.
[40] L. Manevitz, M. Yousef, One-class document classification via neural networks, Neurocomputing 70 (7–9) (2007) 1466–1481.
[41] Dan Pelleg, Andrew W. Moore, Accelerating exact k-means algorithms with geometric reasoning, in: KDD, 1999, pp. 277–281.
[42] G. Ratsch, S. Mika, B. Scholkopf, K. Muller, Constructing boosting algorithms from svms: an application to one-class classification, IEEE Trans. Pattern Anal. Machine Intell. 24 (9) (2002) 1184–1199.
[43] T. Hitendra Sarma, P. Viswanath, B. Eswara Reddy, A hybrid approach to speed-up the k-means clustering method, Int. J. Machine Learn. Cybernet. 4 (2) (2013) 107–117.
[44] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Adaptive computation and machine learning, 2002.
[45] S. Sharma, C. Bellinger, N. Japkowicz, Clustering based one-class classification for compliance verification of the comprehensive nuclear-test-ban treaty, LNAI of Lecture Notes in Computer Science, vol. 7310, 2012, pp. 181–193.
[46] SIAM, Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011(April), SIAM OmniPress, Mesa, Arizona, USA, 2011, pp. 28–30. <http://tunedit.org/challenge/QSAR>
[47] S. Sonnenburg, G. Rtsch, C. Schfer, B. Schlkopf, Large scale multiple kernel learning, J. Machine Learn. Res. 7 (2006) 1531–1565.
[48] A. Szczurek, B. Krawczyk, M. Maciejewska, Vocs classification based on the committee of classifiers coupled with single sensor signals, Chemometr. Intell. Lab. Syst. 125 (2013) 1–10.
[49] R. Tadeusiewicz, M.R. Ogiela, P.S. Szczepaniak, Notes on a linguistic description as the basis for automatic image understanding, Int. J. Appl. Math. Comput. Sci. 19 (1) (2009) 143–150.
[50] D.M.J. Tax, R.P.W. Duin, Support vector data description, Machine Learn. 54 (1) (2004) 45–66.
[51] David M.J. Tax, Robert P.W. Duin, Combining one-class classifiers, in: Proceedings of the Second International Workshop on Multiple Classifier Systems, MCS '01, Springer-Verlag, London, UK, 2001, pp. 299–308.
[52] David M.J. Tax, Piotr Juszczak, Elzbieta Pekalska, Robert P.W. Duin, Outlier detection using ball descriptions with adjustable metric, in: Proceedings of the 2006 Joint IAPR International Conference on Structural, Syntactic, and Statistical Pattern Recognition, SSPR'06/SPR'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 587–595.
[53] D.M.J. Tax, Robert P.W. Duin, Characterizing one-class datasets, in: Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa, 2005, pp. 21–26.
[54] O. Taylor, J. MacIntyre, Adaptive local fusion systems for novelty detection and diagnostics in condition monitoring, Proc. SPIE – Int. Soc. Opt. Eng. 3376 (1998) 210–218.
[55] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2008.
[56] M. van Erp, L. Vuurpijl, L. Schomaker, An overview and comparison of voting methods for pattern recognition, in: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition, 2002, pp. 195–200.
[57] Xizhao Wang, Yadong Wang, Lijuan Wang, Improving fuzzy c-means clustering based on feature-weight learning, Pattern Recogn. Lett. 25 (10) (2004) 1123–1132.
[58] T. Wilk, M. Woźniak, Soft computing methods applied to combination of one-class classifiers, Neurocomputing 75 (January 2012) 185–193.
[59] T. Windeatt, R. Ghaderi, Coding and decoding strategies for multi-class learning problems, Inform. Fusion 4 (1) (2003) 11–21.
[60] M. Woźniak, M. Grana, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inform. Fusion 16 (2014) 3–17.

[61] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Syst, Man Cybernet. 22 (3) (1992) 418–435.
[62] C. Yeh, Z. Lee, S. Lee, Boosting one-class support vector machines for multi-class classification, Appl. Artif. Intell. 23 (4) (2009) 297–315.
[63] H. Yu, J. Man, K.C. Chang, Pebl: Web page classification without negative examples, IEEE Trans. Knowl. Data Eng. 16 (1) (2004) 70–81.
[64] L. Zhang, W. Zhou, L. Jiao, Kernel clustering algorithm, Jisuanji Xuebao/Chinese J. Comput. 25 (6) (2002) 587–590.
[65] H. Zuo, O. Wu, W. Hu, B. Xu, Recognition of blue movies by fusion of audio and video, in: Proceedings of 2008 IEEE International Conference on Multimedia and Expo, ICME 2008, 2008, pp. 37–40.