Journal of Computational Science 9 (2015) 19–25

Contents lists available at ScienceDirect

Journal of Computational Science

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

Incremental weighted one-class classifier for mining stationary data streams

Bartosz Krawczyk*, Michał Woźniak

Department of Systems and Computer Networks, Wrocław University of Technology, Wrocław, Poland

ARTICLE INFO

Article history:
Available online 18 April 2015

Keywords:
Data stream classification
Pattern classification
One-class classification
Incremental learning
Machine learning

ABSTRACT

Big data analytics, especially data stream mining, is among the most popular contemporary machine learning problems. More and more often real-life tasks could generate massive and continuous amounts of data. Standard classifiers cannot cope with a large volume of the training set and/or changing nature of the environment. In this paper, we deal with a problem of continuously arriving objects, that with each time interval may contribute new, useful knowledge to the patter classification system. One-class classification is a very useful tool for stream analysis, as it can be used for tackling outliers, noise, appearance of new classes or imbalanced data to name a few. We propose a novel version of incremental One-Class Support Vector Machine, that assigns weights to each object according to its level of significance. This allows to train more robust one-class classifiers on incremental streams. We present two schemes for estimating weights for new, incoming data and examine their usefulness on a number of benchmark datasets. We also analyze time and memory requirements of our method. Results of experimental investigations prove, that our method can achieve better one-class recognition quality than algorithms used so far.

1. Introduction

Contemporary computer systems store and process enormous amounts of data. Current predictions point out, that the volume of stored information will be doubling every two years. E-mails, social webs, on-line shopping etc. produce evergrowing data, that may carry valuable hidden information. Therefore, three main issues in big data analytics must be addressed (known as 3Vs: *Volume, Velocity, Variety*): how to efficiently transfer such volumes [8], how to store them and how to extract meaningful knowledge from them [14,16]. In this work we are mainly focusing on the *Velocity*, because designing big data analytical tools must take into consideration that most of data in modern systems arrives continuously [3] as so-called data stream [5]. Furthermore, the nature and characteristics of data, i.e., statistical dependencies characterizing an analyzed task may change [9]. A special analytical model which can cope with and adapt to such non-stationary characteristics is required in such cases [6,7,15].

In our research we focus on a stationary data stream classification by one-class classifier (OCC), which may be seen as a specific case of binary classification. It creates a discriminatory model, which is able to assign the available objects coming from the target distribution and unknown outliers. OCC is an attractive approach for data stream classification [4,10], because it can be used for binary classification without an access to counterexamples, decomposing a multi-class data stream, outlier detection or novel class recognition. Most of data stream classification or prediction methods focus on supervised learning, that require a fully labeled dataset for training. However, labeling an entire chunk of data stream is expensive or labels are hard to obtain (e.g., because correct labels can be available with a long time as in the case of credit assessment or most of medical diagnostic cases), which limits the real-life applications of such methods. Furthermore, OCCs require only a small number of positively labeled examples for initial training, which is a valuable property in case of non-stationary classification.

So far the OCC problem for data streams, especially in the presence of concept drift, has not been investigated thoroughly. One should mention several proposal of on-line One-Class Support Vector Machines [18], an one-class modification of very fast decision trees (OcVFDT) [11] and uncertain one-class classifier for summarizing concepts in the presence of uncertainty applying a generated bound score into a one-class SVM-based learning phase [12]. Few ensemble techniques, based on chunks of data and standard one-class classifiers have also been recently introduced [17,19].

In this paper, we propose a novel algorithm for learning an incremental one-class classifier suitable for mining streaming data. It

* Corresponding author.
E-mail addresses: bartosz.krawczyk@pwr.edu.pl (B. Krawczyk), michal.wozniak@pwr.edu.pl (M. Woźniak).

http://dx.doi.org/10.1016/j.jocs.2015.04.024
1877-7503/

is a flexible extension of incremental One-Class Support Vector Machine algorithm, that can be applied to any of already existing solutions of this learning problem. We propose to utilize a weighted version of this classifier and adapt it to the incremental learning problem. Weights are assigned to each object and control the influence it has on the shape of the decision boundary. With this, we are able to more efficiently exploit new, incoming data. Our method allows to assign higher weights to objects that carry useful information that extends the competence of the learner, while delegating low weights to irrelevant, recurrent or noisy examples. We introduce two schemes for calculating new weights for incoming objects from data streams and inputing those weights into one-class incremental learning procedure. We carry experimental study on a number of benchmark and compare our proposal with state-of-the-art algorithms for one-class data stream analysis.

2. Incremental Weighted One-Class Support Vector Machine

One-Class Support Vector Machine (OCSVM) [13] is considered among the most popular and most efficient one-class classifiers. It computes a closed boundary in a form of a hypersphere enclosing all the objects from the target class ω_T. Object belongs to the target class, if it falls within this hypersphere. Otherwise it belongs to outliers.

OCSVM's hypersphere can be sufficiently described by two parameters: center a and a radius R. To have a low acceptance of the possible outliers the volume of this d-dimensional hypersphere, which is proportional to R^d, should be minimized to encompass all of the target class objects without any additional unoccupied decision space. The minimization of R^d implies minimization with respect to R^2. We can formulate the minimization functional as follows:

$$\Theta(a, R) = R^2, \tag{1}$$

with constraint:

$$\forall_{i \in \{1,\ldots,N\}} \quad \|x_i - a\|^2 \leq R^2, \tag{2}$$

where x_i are objects belonging to the target class, and N stands for the number of training objects. Additionally, as in a standard SVM, one may introduce slack variables ξ_i. They allow for some object to lie outside of the hypersphere and can, to some degree, filter out internal noise from the training set.

This idea can be further augmented, creating a Weighted One-Class Support Vector Machine (WOCSVM) [1]. Here, we introduce weights w_i that associate an importance measure to each of the training objects. This forces slack variables ξ_i, to be additionally controlled by w_i. If with object x_i there is associated a small weight w_i then the corresponding slack variable ξ_i indicates a small penalty. In effect, the corresponding slack variable will be larger, allowing x_i to lie further from the center a of the hypersphere. This reduces an impact of x_i on the shape of a decision boundary of WOCSVM.

To apply this, we need to modify the minimization functional:

$$\Theta(a, R) = R^2 + C\sum_{i=1}^{N} w_i \xi_i, \tag{3}$$

with the modified constraints that almost all objects are within the hypersphere:

$$\forall_{i \in \{1,\ldots,N\}} \quad \|x_i - a\|^2 \leq R^2 + \xi_i, \tag{4}$$

where $\xi_i \geq 0$, $0 \leq w_i \leq 1$. C denotes a parameter that controls the optimization process – the larger C, the less outliers are allowed with the increase of the volume of the hypersphere.

For establishing weights we may use techniques dedicated to a Weighted Multi-class Support Vector Machines [2]. We propose to use a method based on distance from the center of the hypersphere:

$$w_i = \frac{|x_i - x_{mean}|}{R + \delta}, \tag{5}$$

where $\delta > 0$ is prevents the case of $w_i = 0$. The value of x_{mean} is computed with the usage of all available learning samples.

We make an assumption, that the classified data stream is given in a form of data chunks. During the first iteration, we have at our disposal an initial data set $\mathcal{DS}_0$ that allows to train the first version of the classifier. Then, for each ith iteration, we receive an additional chunk of data denoted as $\mathcal{DS}_i$.

In case of stationary data streams, each incoming chunk of data may extend the knowledge base derived from the dataset. That is why it is crucial to fuse new data with previous ones. At the same time, the incoming data may be redundant or even noisy, and should be carefully examined. One needs an incremental learning approach, that can handle both of these tasks.

We propose to apply the classifier adaptation in a streaming environment by modification of weights assigned to objects from the data set. We introduce an incremental learning procedure, meaning that the data set $\mathcal{DS}$ will consist of all available chunks of data at the given ith moment. The proposed algorithm is summarized in a pseudo-code manner in Algorithm 1.

Algorithm 1. Incremental WOCSVM for mining data streams.

Require: input data stream,
data chunk size,
incremental training procedure()
$i \leftarrow 0$
repeat
collect new data chunk DS_i
calculate the weights of the examples from DS_i according to Eq. (5) or Eq. (6)
classifier ← incremental training procedure (DS_i, weights assigned to objects in DS_i, previous formed model)
$i \leftarrow i + 1$
until end of the input data stream

The proposed algorithm is flexible and can be applied with any of existing incremental OCSVM methods. Proposed procedure adds the weighting and adapting process to incremental learning, without modifying the algorithm itself. For this paper, as a base for modification we use incremental OCSVM proposed in [18].

Incremental learning should allow to change the shape of the previously learned decision boundary. Here, we propose two strategies for updating our classifier with new, incoming data by managing their weights:

- assigning weights to objects from $\mathcal{DS}_k$ according to Eq. (5). This is motivated by the fact that in the incoming data chunk not all objects should have the same impact on the shape of the new decision boundary,
- assigning the highest weights to objects coming from the new data chunks:

$$\forall_{x_i \in \mathcal{DS}_k} \quad w_i = 1. \tag{6}$$

This is motivated by the fact that given chunk represent the current state of the streaming environment and therefore should have a top priority in forming the new decision boundary.

3. Experimental investigations

Our experiments were carried out on stationary data streams, in which we aim at increasing the knowledge of classifier by sequential addition of new data. We examined the usefulness of the two proposed strategies for calculating weights assigned to each object

Table 1
Details of datasets used in the experimental investigation. Numbers in parentheses indicates the number of objects in the minority class.

Name	Objects	Features	Classes
Breast-Wisconsin	699 (241)	9	2
Pima	768 (268)	8	2
Yeast3	1484 (163)	8	2
Voting records	435 (168)	16	2
CYP2C19 isoform	837 (181)	242	2
RBF	1 000 000	20	2

Table 2
Average classification accuracies [%].

Data set	OcVFDT	mIOCSVM	L_1F_0	L_2F_0
Breast-Wisconsin	67.73	73.81	**75.98**	71.51
Pima	71.39	72.40	**72.90**	71.30
Yeast3	70.24	70.90	**71.94**	69.83
Voting records	75.03	77.56	**78.87**	76.50
CYP2C19 isoform	68.71	72.85	**75.00**	70.81
RBF	73.98	74.81	**81.03**	78.51

The bolded values stand for the highest obtained accuracy.

in the incremental OCSVM learning, and we compared them to state-of-the-art methods for one-class data stream mining.

3.1. Data sets

There is an abundance of benchmarks for comparing machine learning algorithms working in static environments. For creating stationary data streams, we use popular benchmarks and divide them into chunks of objects in each, given sequentially to the classifier.

Additionally, up to authors best knowledge, there are no data stream benchmarks for one-class classification problems. Therefore, we need to apply a transformation changing existing multi-class streams into one-class data. To do this, we need to select a single class to serve as the target concept and use remaining class(es) as outliers. Training is performed only on target class objects, while testing on both target and outliers representatives.

Classifier is trained on given chunk and tested on incoming one. Then we fuse incoming chunk with the existing one and repeat the procedure for new data.

Details of used benchmarks for stationary data streams can be found in Table 1.

3.2. Set-up

For the purposes of experimental analysis, we use as a base model WOCSVM with RBF kernel, $\sigma = 0.1$ and $C = 10$.

Table 3
Average chunk training time in seconds [s].

Data set	OcVFDT	IOCSVM	L_1F_0	L_2F_0
Breast-Wisconsin	1.02	4.12	4.56	4.23
Pima	1.38	4.17	4.62	4.28
Yeast3	1.02	4.02	4.14	4.05
Voting records	1.01	3.78	4.02	3.89
CYP2C19 isoform	0.97	4.06	4.69	4.18
RBF	0.76	2.45	3.04	2.53

Table 4
Average memory consumption in megabytes [MB].

Data set	OcVFDT	IOCSVM	L_1F_0	L_2F_0
Breast-Wisconsin	0.35	1.26	1.44	1.35
Pima	0.28	1.02	1.28	1.13
Yeast3	0.32	1.15	1.30	1.21
Voting records	0.17	0.97	1.10	1.02
CYP2C19 isoform	0.68	1.87	2.03	1.92
RBF	0.73	2.01	2.72	2.64

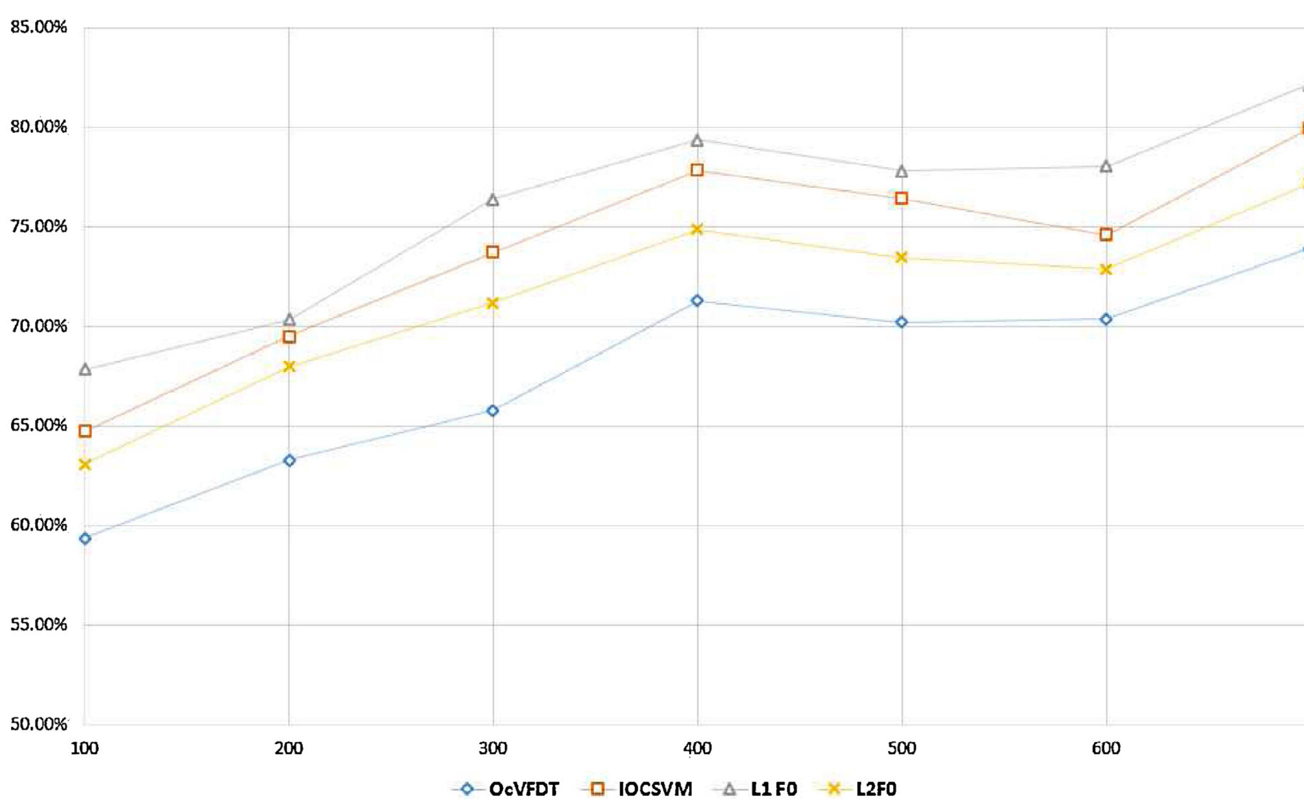

Fig. 1. Accuracies of proposed incremental learning for Breast-Wisconsin data set.

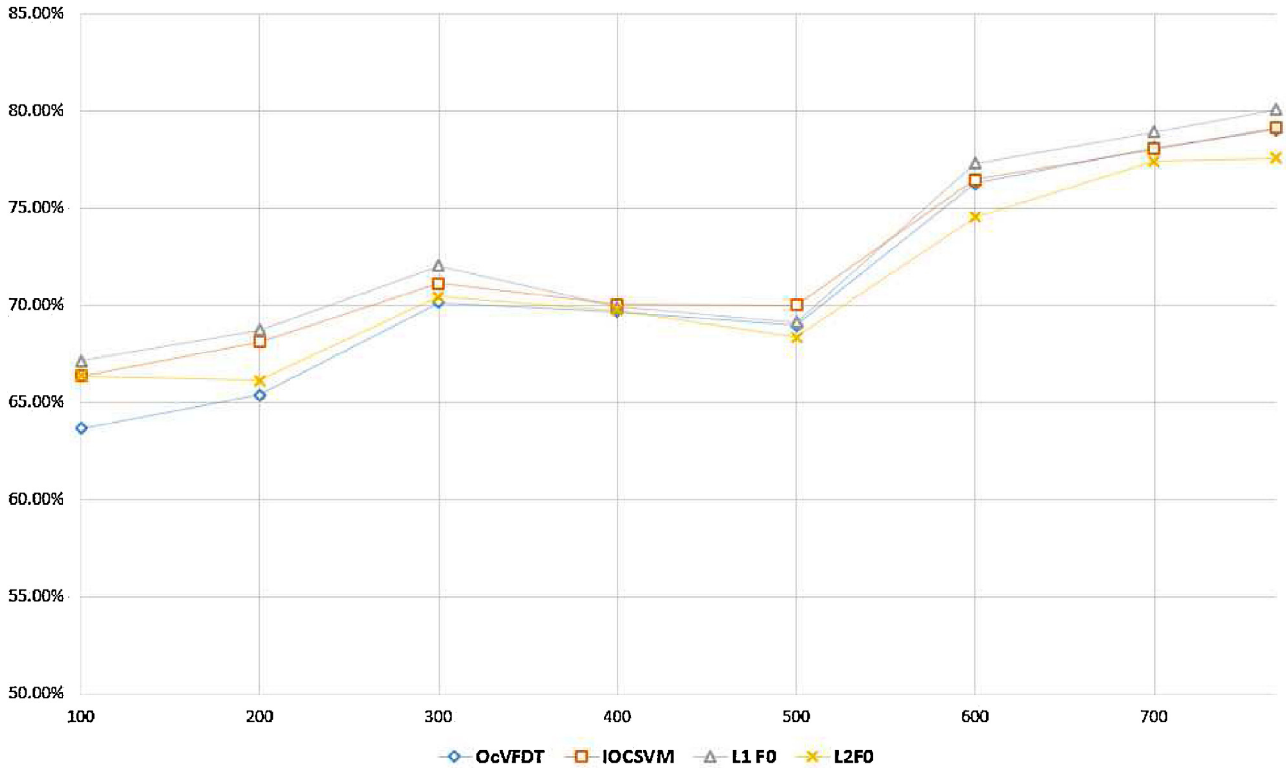

Fig. 2. Accuracies of proposed incremental learning for Pima data set.

We use two different models of the proposed one-class classifiers in our experiments – each with applied different incremental learning mechanism (L_1F_0 stands for assigning weights to new objects according to Eq. (5, while L_2F_0 for assigning the highest weights to new objects). As reference methods, we apply an Incremental and On-line One-Class Support Vector Machine (IOCSVM, with RBF kernel, σ = 0.1 and C = 10) [18] and One-Class Very Fast Decision Tree (OcVFDT) [11].

The data block size used for creating data chunks was d = 100 for first five (small) data sets and d = 2500 for RBF dataset.

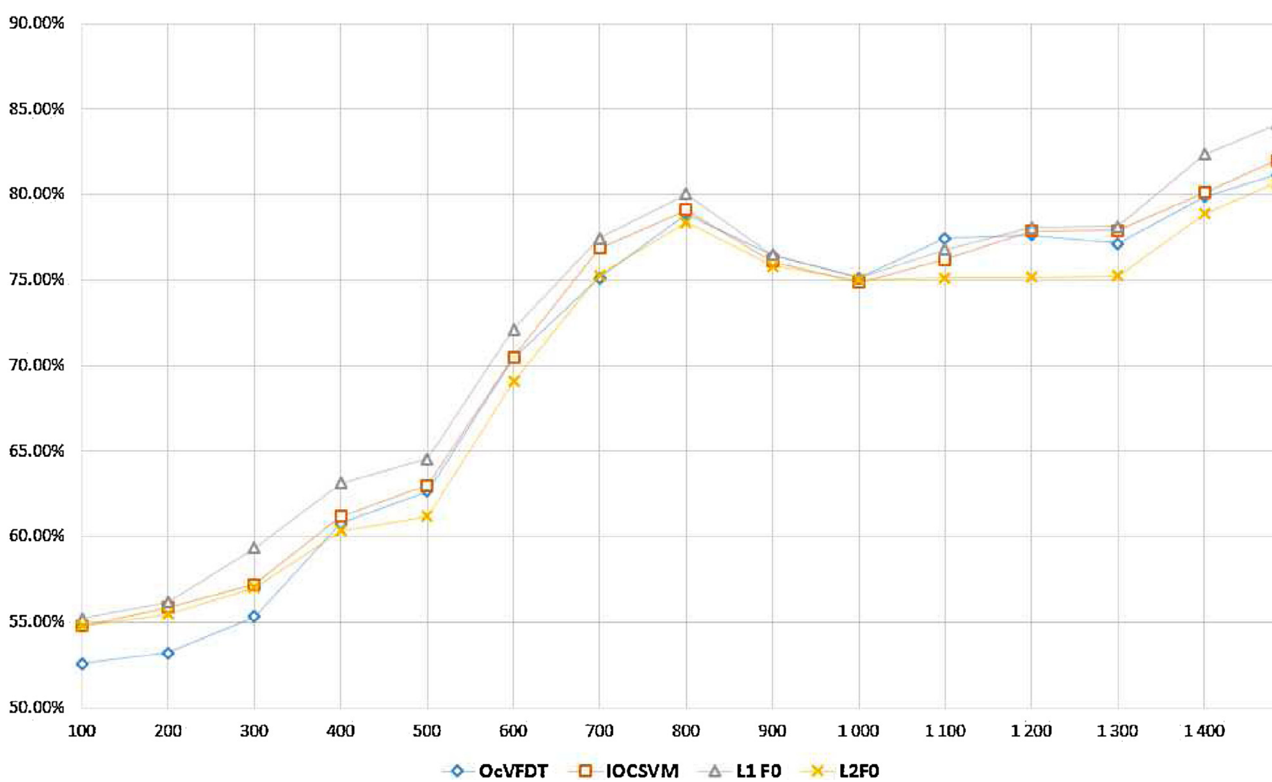

Fig. 3. Accuracies of proposed incremental learning for Yeast3 data set.

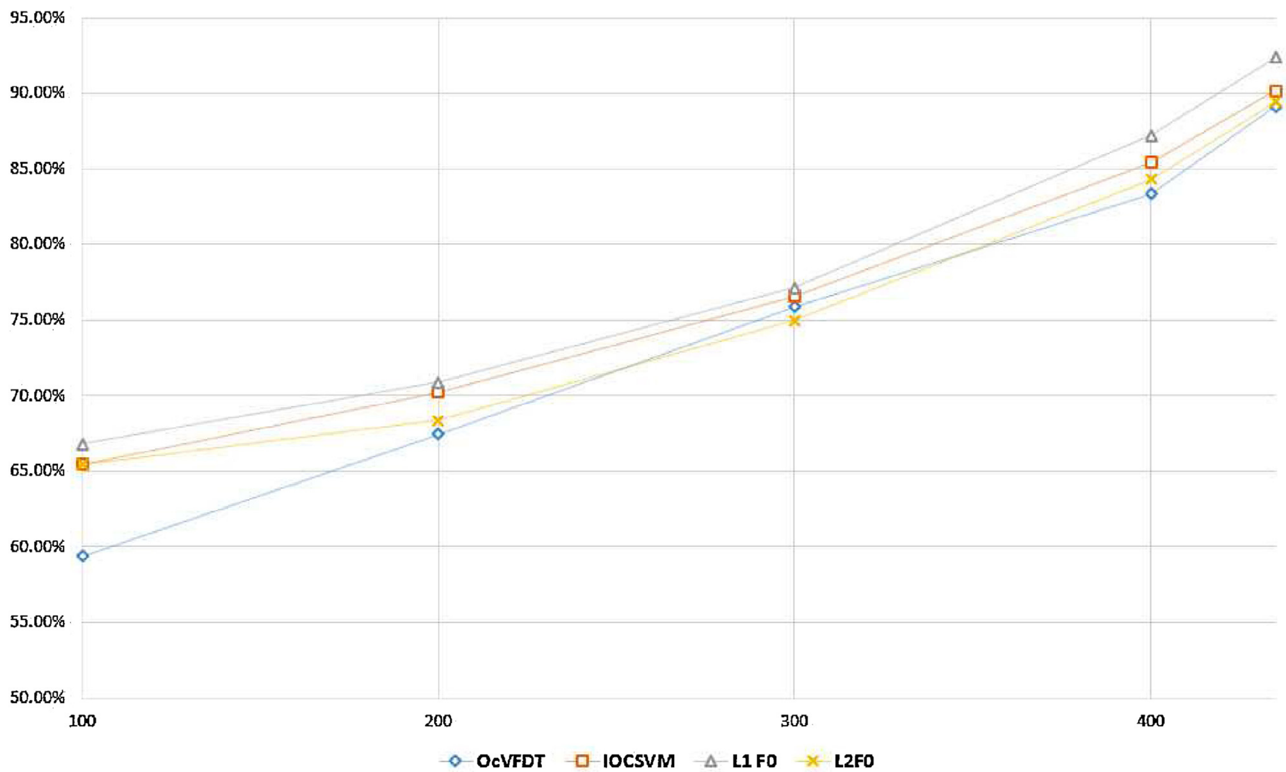

Fig. 4. Accuracies of proposed incremental learning for Voting records data set.

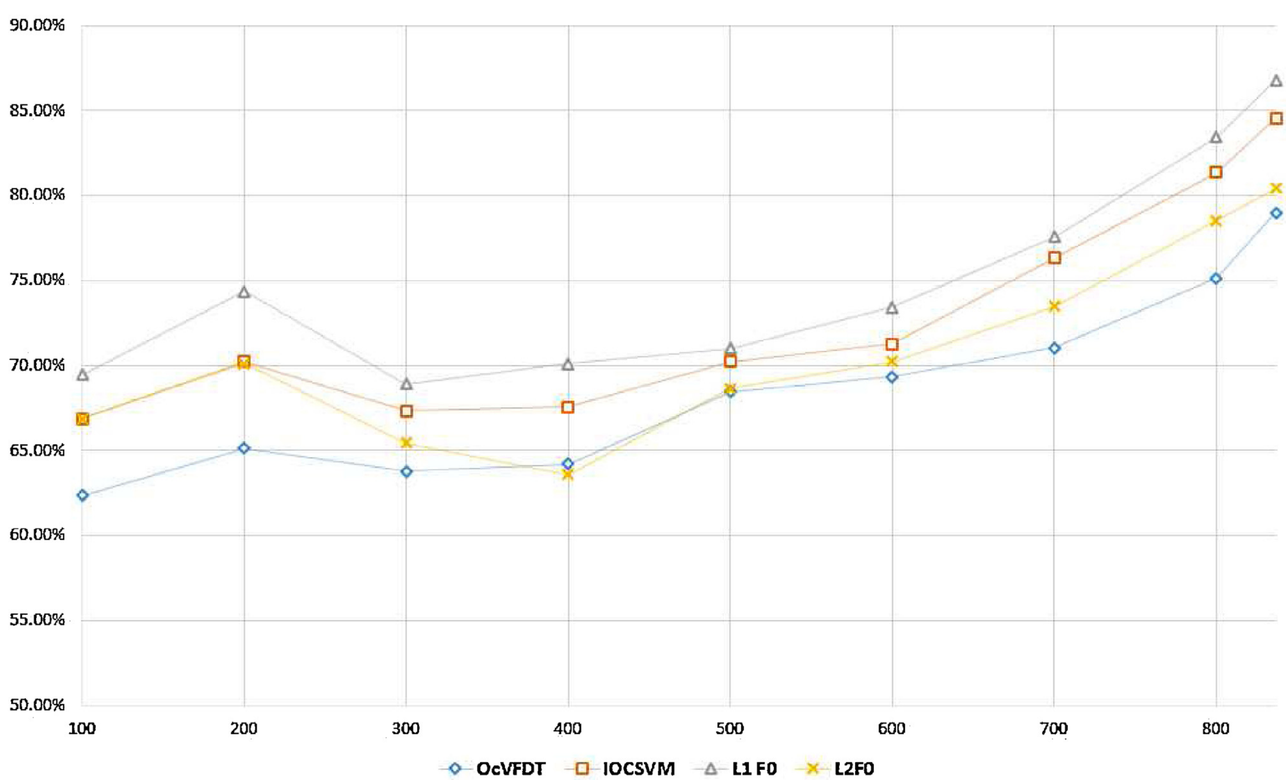

Fig. 5. Accuracies of proposed incremental learning for CYP2C19 isoform data set.

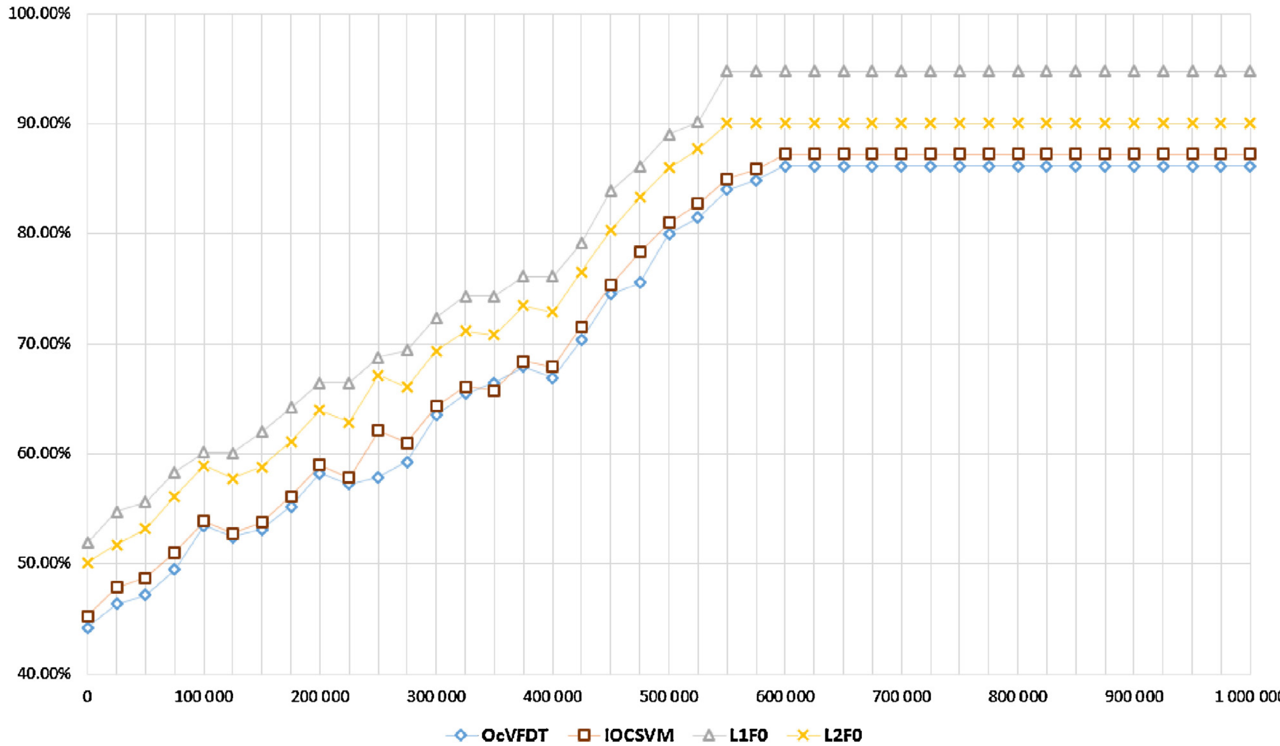

Fig. 6. Accuracies of proposed incremental learning for RBF data set.

3.3. *Results and discussion*

Average accuracies achieved by tested algorithms are presented in Table 2, their average training times on data chunk are presented in Table 3, and their average memory usage is presented in Table 4. Additionally, we present detailed accuracy behavior of the proposed incremental learning and forgetting schemes on each of analyzed data sets in Figs. 1–6 to allow for a visual inspection of their performance.

From the obtained results, we may draw several conclusions. First of all, out of two proposed methods for incremental weight calculation the one based on Eq. (5) is always superior to the method that assigns highest weights to new data. This is because of increased selectiveness of this approach – only new objects that carry new and useful knowledge extending the competence of the classifier are given high priority. In case of second method, all new objects are equally important, even if they are redundant or noisy. Therefore, first scheme is able to more efficiently explore the incoming incremental data stream.

When comparing the proposed method with reference classifiers, we may see that our proposal outperform both one-class decision tree and incremental OCSVM. Decision tree is by nature a binary classifier and require some generalizations in order to work without counterexamples. Due to this it cannot capture the properties of analyzed stream properly. OCSVM is a much more efficient classifier, but we show that with simple proposed modification we are able to significantly boost its classification efficiency.

We can see, that for RBF and Breast-Wisconsin datasets we obtained asymptotic results. This is a desirable property in case of stationary data streams. For the remaining datasets, this property was not observed. This can be explained by the fact, that given data stream was too small to stabilize the model.

Finally, we can see that the proposed upgrade of incremental OCSVM into incremental WOCSVM does not effect in a increase of training time or memory usage. This shows, that with the proposed change in implementation, we are able to improve the performance of the incremental one-class classifiers without increasing their computational load.

4. Conclusions

In this work, we proposed an incremental Weighted One-Class Support Vector Machine for mining streaming data.

Our proposition focused on the modifications of weights used by WOCSVM. We employed the incremental training of the weights in order to efficiently exploit the incoming data chunks. The proposed scheme is highly selective – only new objects that carry new and useful knowledge extending the competence of the classifier influence the shape of the decision boundary. Redundant or noisy examples are filtered out by assigning to them marginal weights.

Experimental analysis and comparison with other methods for one-class data stream mining prove the efficiency of our approach. Not only we are able to increase the classification accuracy in comparison to standard incremental OCSVM, but also achieve this without any additional computational costs.

As the achieved results are very promising, then we decided to continue our work with them in the future. We plan to use our method for data streams with concept drift and presence of class imbalance, and to propose a classifier ensemble based on the proposed incremental WOCSVM.

Acknowledgements

This work was partially supported by the Polish National Science Center under the grant no. DEC-2013/09/B/ST6/02264 and by EC under FP7, Coordination and Support Action, Grant Agreement Number 316097, ENGINE European Research Centre of Network Intelligence for Innovation Enhancement (http://engine.pwr.wroc.pl/).

References

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

[2] B. Cyganek, One-class support vector ensembles for image segmentation and classification, J. Math. Imaging Vis. 42 (2–3) (2012) 103–117.

[3] B. Cyganek, S. Gruszczyński, Hybrid computer vision system for drivers' eye recognition and fatigue monitoring, Neurocomputing 126 (2014) 78–94.

[4] I. Czarnowski, P. Jedrzejowicz, Ensemble classifier for mining data streams, in: 18th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES 2014, Gdynia, Poland, 15–17 September 2014, 2014, pp. 397–406.

[5] J. Gama, A survey on learning from data streams: current and future trends, Prog. AI 1 (1) (2012) 45–55.

[6] K. Jackowski, Fixed-size ensemble classifier system evolutionarily adapted to a recurring context with an unlimited pool of classifiers, Pattern Anal. Appl. 17 (4) (2014) 709–724.

[7] D. Jankowski, K. Jackowski, Evolutionary algorithm for decision tree induction, in: Computer Information Systems and Industrial Management – 13th IFIP TC8 International Conference, CISIM 2014, Ho Chi Minh City, Vietnam, 5–7 November 2014. Proceedings, 2014, pp. 23–32.

[8] W. Kmiecik, K. Walkowiak, Metaheuristic algorithms for optimization of survivable multicast overlay in dual homing networks, Cybernet. Syst. 44 (6–7) (2013) 606–626.

[9] B. Krawczyk, J. Stefanowski, M. Woźniak, Data stream classification and big data analytics, Neurocomputing 150 (2015) 238–239.

[10] B. Krawczyk, M. Woźniak, Incremental learning and forgetting in one-class classifiers for data streams, in: R. Burduk, K. Jackowski, M. Kurzyński, M. Woźniak, A. Żołnierek (Eds.), Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013, Milkow, Poland, 27–29 May 2013, Springer International Publishing, 2013, pp. 319–328, volume 226 of Advances in Intelligent Systems and Computing.

[11] C. Li, Y. Zhang, X. Li, Ocvfdt: one-class very fast decision tree for one-class classification of data streams, in: Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data, SensorKDD'09, ACM, 2009, pp. 79–86.

[12] B. Liu, Y. Xiao, P.S. Yu, L. Cao, Y. Zhang, Z. Hao, Uncertain one-class learning and concept summarization learning on uncertain data streams, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 468–484.

[13] B. Schölkopf, A.J. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond, in: Adaptive Computation and Machine Learning, MIT Press, 2002.

[14] I. Triguero, D. Peralta, J. Bacardit, S. García, F. Herrera, MRPR: a mapreduce solution for prototype reduction in big data classification, Neurocomputing 150 (2015) 331–345.

[15] M. Woźniak, A hybrid decision tree training method using data streams, Knowl. Inf. Syst. 29 (2) (2011) 335–347.

[16] M. Woźniak, P. Cal, B. Cyganek, The influence of a classifiers diversity on the quality of weighted aging ensemble, in: N.T. Nguyen, B. Attachoo, B. Trawinski, K. Somboonviwat (Eds.), Intelligent Information and Database Systems, volume 8398 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 90–99.

[17] D. Zhang, L. Cai, Y. Wang, L. Zhang, A learning algorithm for one-class data stream classification based on ensemble classifier, in: Computer Application and System Modeling (ICCASM), 2010 International Conference on volume 2, 2010, pp. V2-596–V2-600.

[18] Y. Zhang, N. Meratnia, P. Havinga, Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks., in: International Conference on Advanced Information Networking and Applications Workshops, 2009, WAINA'09, 2009, pp. 990–995.

[19] X. Zhu, W. Ding, P.S. Yu, C. Zhang, One-class learning and concept summarization for data streams, Knowl. Inf. Syst. 28 (3) (2011) 523–553.