ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

One-class classification algorithm based on convex hull

Diego Fernandez-Francos, Oscar Fontenla-Romero and Amparo Alonso-Betanzos *

Department of Computer Science, University of A Coruña, Spain

Abstract. A new version of a one-class classification algorithm is presented in this paper. In it, convex hull (CH) is used to define the boundary of the target class defining the one-class problem. An approximation of the D-dimensional CH decision is made by using random projections and an ensemble of models in very low-dimensional spaces. Expansion and reduction of the CH model prevents over-fitting. So a different method to obtain the expanded polytope is proposed in order to avoid some undesirable behavior detected in the original algorithm in certain situations. Besides, this modification allows the use of a new parameter, the CH center, that provides even more flexibility to our proposal. Experimental results showed that the new algorithm is significantly better, regarding accuracy, than the previous work on a large number of datasets.

1 Introduction

One of the problems in pattern recognition is to classify some objects into classes according to their features. A particular case, known as one-class classification [1], is a binary classification task for which only information of one class (target class) is available for learning. This means that the classifier does not need any assumption on the outlier data to estimate the decision boundary. The geometrical structure of the convex hull (CH) has been used to define the class boundaries in multiclass [2, 3, 4] and one-class classification problems [5]. The use of conventional implementations of the CH in high dimensions, due to its high computational complexity, is not feasible. New implementations have been proposed to deal with this problem [6, 7]. In this paper, we propose a new version of a one-class classification algorithm presented by Casale et al. [5], that approximates the D-dimensional CH decision by means of random projections and an ensemble of CH models in 2 dimensions which can be used for high dimensions in an acceptable execution time. A new formula to calculate a expanded version of the CH that prevents over-fitting was proposed in order to avoid an undesirable behavior detected in the original algorithm: the appearance of non-convex polytopes. Besides, using this formula a new parameter (the center of the polytope) can be used, providing even more flexibility to our algorithm.

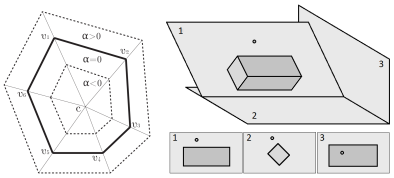
*This work has been supported by the *Secretaría de Estado de Investigación* of the Spanish Government (Grants TIN2012-37954 and TIN2015-65069-C2-1-R). Diego Fernandez Francos would like to thank the *Ministerio de Ciencia e Innovación* for FPI supporting grant.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

2 Background

This section presents the main characteristics of the original one-class algorithm proposed by Casale et al.[5]. The main contributions of this work in the context of one-class classification were: 1) the use of the geometric structure of the convex hull (CH) to model the boundary of the target class defining the one-class problem, 2) the use of reduced and enlarged versions of the original CH, called *extended convex polytopes* (ECP), in order to avoid over-fitting and to find the optimal operating point of the classifier, and 3) the decision whether a point belongs to the D-dimensional ECP model was made by using an ensemble of decisions in very low-dimensional spaces $d \ll D$ ($d = 1$ or $d = 2$). This was called *approximate convex polytope decision ensemble* (APE).

The CH provides a tight approximation among several convex forms to the class region of a set of points $S \subseteq R^n$. However, this approximation is prone to over-fitting. To avoid this, reduced/enlarged versions of the original CH were used. Vertices of this ECP are defined with respect to the center point $c = \frac{1}{N}\sum_i x_i, \forall x_i \in S$ and the expansion parameter α as in $v_\alpha : \left\{ v + \alpha \frac{(v-c)}{\|v-c\|} \middle| v \in Conv(S) \right\}$. Fig. 1(a) shows a reduced (inner dashed polygon) and enlarged (outer dashed polygon) ECP. Calculating the ECP and testing if a



(a) (b)

Fig. 1: (a) Reduced/enlarged extended convex polytope. (b) APE strategy.

point lies inside it in high dimensional spaces is computationally unfeasible. To overcome this limitation, the APE was proposed. It consists in approximating the decision made by the ECP in the original D-dimensional space by means of a set of τ randomly projected decisions made on low-dimensional spaces. In this scenario, the *decision rule* is the following: *a point does not belong to the modeled class if and only if there exists at least one projection in which the point lies outside the projected convex polytope.* Fig. 1(b) shows a three-dimensional convex polytope approximated by three random projections in 2D. As can be seen, a point that lies outside the original polytope might appear inside one or more projections.

2.1 Expansion factor in the low dimensional space

The decision that a point belongs to the target class is made by considering the ECP in a low-dimensional space. Since the projection matrix is randomly generated, the norm of the original space is not preserved in the resulting one. Thus, a constant value of α in the original space corresponds to a set of values γ_i in the projected one. So, the set of vertices that define the low-dimensional

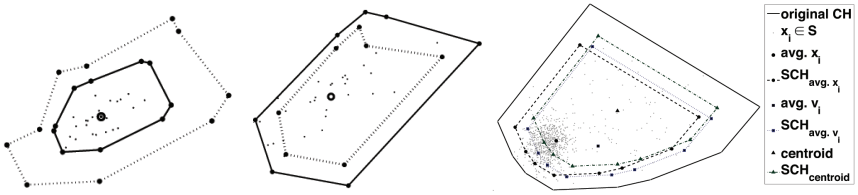
ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

approximation of the expanded polytope are: $\overline{v}^{\alpha} : \left\{ \overline{v_i} + \gamma_i \frac{(\overline{v_i} - \overline{c})}{\|\overline{v_i} - \overline{c}\|} \right\}, i = 1 \dots N$ where $\overline{c} = Pc$ represents the projection of the center c given a random projection matrix P, $\overline{v_i}$ is the set of convex hull vertices of the projected data and γ_i is defined as $\gamma_i = \frac{(v_i - c)^T P^T P (v_i - c)}{\|v_i - c\|} \alpha$ where v_i is the ith vertex of the CH in the original space.

3 Proposed method

In this work we propose some modifications related to the way the ECP is obtained in the previous algorithm (see sec. 2.1). The main purpose is to avoid an awkward behavior detected when the expanded/reduced version of the polytope is obtained and provide, at the same time, a more intuitive expansion parameter and a more flexible method.

As can be seen in section 2.1, the expansion factor used in the original algorithm takes into account the distances between the vertices and the center of the CH in the original space to calculate the ECP. Due to this, each vertex $\overline{v_i}$ in the projected space has its own expansion parameter γ_i. These expansion factors can lead to obtain non-convex polytopes, which is a non desirable behavior of the algorithm, that aims at checking whether a point lies inside of a convex polytope. Figures 2(a) and 2(b) show two examples of this undesirable behavior. Data of class 1 (iris setosa) of the Iris dataset [8] and two different random projections into 2 dimensions were used to get these results.



(a) $\alpha = 0.4$ (b) $\alpha = -0.1$ (c)

Fig. 2: Examples of non-convex polytopes as a result of the (a) expansion or (b) reduction of the projected CH. (c) Decision regions for each type of center.

In this work we propose the use of the Scaled Convex Hull (SCH) presented by Liu et al. [2] to calculate the ECP in the low dimensional space. Vertices are defined with respect to the expansion parameter $\lambda \in [0, +\infty)$ as in $\overline{v}^{\lambda} : \{\lambda \overline{v_i} + (1 - \lambda)\overline{c}\}, i = 1 \dots N$. This is a more intuitive formula that only involves information in the projected space and where all the vertices of the projected CH are expanded by the same factor λ, avoiding the previous problem. Besides, using this alternative, it is possible to employ different definitions of CH "center". In the original method, the average of all the points in the D-dimensional space was the unique alternative as a center (see sec. 2.1). However, in this work we propose three different definitions of center: 1) Average of all the points in the projected space $\overline{c} = \frac{1}{N} \sum_i \overline{x_i}, \forall \overline{x_i} \in \overline{S_t}$, 2) Average of the CH vertices in the projected space $\overline{c} = \frac{1}{N} \sum_i \overline{v_i}, \forall \overline{v_i} \in Conv(\overline{S_t})$,and 3) Centroid [9]. As can be

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

seen in Figure 2(c), each type of center leads to different decision regions ($\lambda = 0.8$ has been used), giving more flexibility to our method. In the experimental section we will test the performance of the algorithm using each type of center.

The proposed learning and testing procedures are described in Algorithms 1 and 2, respectively. In the learning phase, the number of projections τ, the expansion parameter λ, and the type of center used to calculate the ECP have to be defined, while in the original method, vertices of the ECP were obtained in the testing algorithm. Changing this operation from the testing to the learning algorithm, as we propose, reduces the computational time as the computations are done only once. Algorithm 2 takes as inputs: the model M and a test

Algorithm 1 Learning algorithm

Input: Training set $S \in R^D$; Number of projections τ; Expansion parameter λ; Type of center tc.
Output: The model M composed of τ projection matrices and their respective ECP vertices.
1: $M = \phi$;
2: **for** $t = 1..\tau$ **do**
3: $P_t \sim N(0,1)$ % Create a random projection matrix;
4: $\overline{S_t} : \{P_t x | x \in S\}$ % Project original data;
5: $\{v_i\}_t = \text{Conv } \overline{S_t}$ % Return the vertices of the CH;
6: $\overline{c} = getCenter(tc)$ % Return the selected center
7: $v_t^\lambda : \{\lambda v_i + (1-\lambda)\overline{c} | \, v_i \in \{v\}_t\}$ % ECP in the low dimensional space;
8: $M = M \cup (P_t, v_t^\lambda)$ % Store the vertices of the projected CH and the projection matrix;
9: **end for**

point $x \in R^D$. At each iteration t, the test point is projected into the low dimensional space spanned by the t-th projection matrix. Then, given the set

Algorithm 2 Testing algorithm

Input: Test point $x \in R^D$; Model M; Parameter α.
Output: $Result \in \{INSIDE, OUTSIDE\}$
1: Result = INSIDE;
2: **for** $t = 1..\tau$ **do**
3: $\overline{x_t} : \{P_t x\}$ % Project test point;
4: **if** $\overline{x_t} \notin$ Conv v_t^λ **then**
5: Result = OUTSIDE;
6: Break;
7: **end if**
8: **end for**

of expanded vertices, it is possible to check whether the point lies inside the projected polytope.

4 Experimental results

In this section, both the proposed SCH algorithm and the original one (APE-2)[5] were tested on 28 one-class problems derived from 11 UCI machine learning repository datasets [8] (see Table 1). For each dataset, one-class problems were obtained considering one of the classes as target and the rest as outliers. In the second column of Table 2, the class considered as target and the number of samples of the target and outlier classes are displayed, respectively. Each problem was evaluated using 10-fold stratified cross-validation on 10 different

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

permutations of the data. Both algorithms were used projecting data down to 2 dimensional spaces and the number of projections was arbitrarily set to 100. In [5] it was shown that taking 2D projections provides better results with a lower number of projections than using 1D projections. Results obtained are reported

Dataset	No. features	No. instances	No. classes
Balance	4	625	3
Breast	10	683	2
Car	6	1728	4
Glass	10	214	3
Haberman	3	306	2
Ionosphere	34	351	2
Iris	4	150	3
Pima	8	768	2
Sonar	60	208	2
TicTacToe	9	958	2
Wine	13	178	3

Table 1: Characteristics of the datasets.

in Table 2. Each element of the table shows the mean Area Under the Rock Curve (AUC) and the standard deviation (SD) obtained. In order to compute the AUC, each curve was evaluated on 100 point varying the α parameter in APE-2 and the λ parameter in SCH approaches. A pairwise t-test [10] between

Dataset	Class	APE-2	SCH$_{avg.x_i}$	SCH$_{avg.v_i}$	SCH$_{centroid}$
Balance	1 (288,337)	89.83±2.2	**90.21±2.5**	90.01±2.7	90.18±2.1
	2 (49,576)	84.34±4.4	**86.38±3.5***	85.06±4.7	85.04±3.9
	3 (288,337)	89.45±2.7	89.70±2.6	89.49±2.7	**89.77±2.6**
Breast	1 (444,239)	95.14±2.6	**95.43±2.2**	95.24±2.5	95.34±2.4
	2 (239,444)	85.01±5.9	85.37±5.4	85.44±5.2	**85.61±4.9**
Car	1 (1210,518)	71.81±3.2	71.74±3.7	**72.00±4.5**	71.57±4.0
	2 (384,1344)	94.76±1.7	95.38±1.5	95.16±1.5	**95.48±1.5***
	3 (69,1659)	96.29±4.9	97.93±3.3	98.63±2.6	**98.76±2.5***
	4 (65,1663)	99.24±2.2	**99.75±0.8***	99.61±1.4	99.72±0.8
Glass	1 (70,144)	98.51±3.7	**98.58±2.9**	98.57±2.8	98.38±3.1
	2 (76,138)	**93.39±5.9***	91.09±5.9	91.29±6.2	91.13±6.3
	3 (68,146)	**95.95±4.4***	94.37±6.0	94.22±5.9	93.97±6.7
Haberman	1 (225,81)	**53.91±7.9**	53.23±9.8	53.73±11.5	53.63±9.8
	2 (81,225)	56.36±8.8	57.14±7.9	**57.86±7.2**	57.65±7.7
Ionosphere	1 (225,126)	90.32±4.5	**91.23±2.7**	91.16±2.9	91.04±2.4
	2 (126,225)	50.12±1.7	50.25±1.3	**50.30±1.4**	50.11±1.3
Iris	1 (50,100)	100±0.0	100±0.0	100±0.0	100±0.0
	2 (50,100)	92.64±6.6	93.92±5.7	94.02±5.4	**94.25±6.5***
	3 (50,100)	92.67±7.7	92.53±7.5	**92.85±7.9**	92.65±6.4
Pima	1 (500,268)	62.73±2.9	**62.85±2.2**	62.60±2.5	62.42±2.4
	2 (268,500)	54.45±3.2	54.41±3.6	**54.84±3.5**	54.52±4.5
Sonar	1 (97,111)	59.90±6.1	61.50±6.2	62.04±4.9	**62.27±4.7***
	2 (111,97)	72.90±5.3	74.53±5.2	**74.74±4.9***	74.07±3.6
TicTacToe	1 (626,332)	59.74±5.0	60.23±4.5	60.32±4.7	**60.36±5.3**
	2 (332,626)	**61.10±5.3**	60.09±5.3	60.74±5.2	61.01±4.6
Wine	1 (59,119)	95.93±5.2	**97.54±3.5***	97.20±4.1	97.52±4.2
	2 (71,107)	81.54±9.6	**82.65±9.2**	81.94±9.8	82.49±9.1
	3 (48,130)	95.04±6.8	96.82±5.1	97.44±3.6	**97.73±3.2***

Table 2: AUC and SD. In bold font, the best result for each problem.

the best result of the SCH approaches and the result obtained by the APE was applied to evaluate the statistical difference at 95% significance level. Methods that are statistically different are marked with an *. SCH versions, varying

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

the "center" used to compute the extended CH (see sec. 3), generally achieve better results than APE-2. Nine times against two, one of the SCH versions is statistically better than the APE-2. As can be seen in the results table, none of the SCH approaches is consistently better than the others. So, depending on the problem at hand (the nature of the data), different "centers" may be used, obtaining dissimilar decision regions that can better fit the data.

5 Conclusions

In this work, a modified version of the approximate polytope ensemble algorithm was presented. A different formula to calculate the expanded polytope was proposed in order to avoid an undesirable behavior detected in the original algorithm, the appearance of non-convex polytopes. Besides, this modification allowed the use of a new parameter (the center of the polytope), that provides more flexibility to our algorithm. Three different ways of calculating the "center" of the polytope were proposed. This modifications were validated in 28 one-class problems. Experimental results demonstrated that the modified algorithm improves the already good performance of the original proposal. Furthermore, results showed that the choice of centers may affect considerably classification results for a given problem.

Future lines of research are a new version of this method able to tackle distributed data and a highly parallel implementation on Apache Spark.

References

[1] Shehroz S Khan and Michael G Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(03):345–374, 2014.

[2] Zhenbing Liu, JG Liu, Chao Pan, and Guoyou Wang. A novel geometric approach to binary classification based on scaled convex hulls. *Neural Networks, IEEE Transactions on*, 20(7):1215–1220, 2009.

[3] Ralph L Kodell, Chuanlei Zhang, Eric R Siegel, and Radhakrishnan Nagarajan. Selective voting in convex-hull ensembles improves classification accuracy. *Artificial intelligence in medicine*, 54(3):171–179, 2012.

[4] Ming Zeng, Yu Yang, Jinde Zheng, and Junsheng Cheng. Maximum margin classification based on flexible convex hulls for fault diagnosis of roller bearings. *Mechanical Systems and Signal Processing*, 66:533–545, 2016.

[5] Pierluigi Casale, Oriol Pujol, and Petia Radeva. Approximate polytope ensemble for one-class classification. *Pattern Recognition*, 47(2):854–864, 2014.

[6] Antonio Ruano, Hamid Reza Khosravani, and Pedro M Ferreira. A randomized approximation convex hull algorithm for high dimensions. *IFAC-PapersOnLine*, 48(10):123–128, 2015.

[7] Min Tang, Jie-yi Zhao, Ruo-feng Tong, and Dinesh Manocha. Gpu accelerated convex hull computation. *Computers & Graphics*, 36(5):498–506, 2012.

[8] M Lichman. Uci machine learning repository (2013). *http://archive. ics. uci. edu/ml*.

[9] Gerard Bashein and Paul R Detmer. Centroid of a polygon. In *Graphics gems IV*, pages 3–6. Academic Press Professional, Inc., 1994.

[10] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.