Ensemble probability distribution for novelty detection

Xiaoshuang Qiao[1,*], *Hui* Wang[2], *Gongde* Guo[1], and *Yuanyuan* Liu[1]

[1]Digit Fujian Internet-of-Things Laboratory of Environmental Monitoring, School of Mathematics and Information，Fujian Normal University, Fuzhou，P.R. China
[2]School of Computing and Mathematics University of Ulster at Jordanstown, Northern Ireland, UK

Abstract. This paper explores a new ensemble approach called Ensemble Probability Distribution Novelty Detection (EPDND) for novelty detection. The proposed ensemble approach provides a metric to characterize different classes. Experimental results on 4 real-world datasets show that EPDND exhibits competitive overall performance to the other two common novelty detection approaches – Support Vector Domain Description and Gaussian Mixed Models in terms of accuracy, recall and F1 scores in many cases.

1 Introduction

One of the basic assumptions in most supervised machine learning algorithms is that the class label set is predefined and shared by the training and testing sets so that the classification model could have a good generalization capability. However, there are so many cases in open-domain applications make this assumption invalide. For example, in online webpage classification, we can easily list out some common classes, such as entertainment, politics, and sports. But it is extremely difficult to provide a complete list of all classes beforehand due to the webpages on any new topics and new classes can appear as the data comes. The classification performance will be degraded sharply while the new classes that are never defined in the training phase emerge in the testing phase. Similar examples can be found in many domains, such as fraud detection, ecosystem disturbance, and so on. Novelty detection is a challenging problem we need to explore and research. Novelty detection is defined as the task of recognising that test data differ in some respects from the data that are used in training stage [1][2].

In this paper, we propose an efficient ensemble framework to detect novelty and present a specialization of the framework involving 5 individual classifiers. The rest of the paper is organized as follows. Section 2 briefly reviews the related work. Section 3 describes ensemble learning and confidence distribution. Section 4 describes the proposed method for novelty detection in details followed by experiments and related analysis in section 5. Finally, section 6 concludes the paper.

2 Related work

* Corresponding author: 815579916@qq.com

For most applications, the acquisition of novelties becomes serious obstacle. Many approaches are proposed for novelty detection. According to the recent work, the widely used techniques in novelty detection mainly consist of the following types: probabilistic, nearest neighbour-based, domain-based approaches, clustering-based.[1]

Probability approaches are based on estimating the generative probability density function (PDF) of the data. The resultant distribution may then be threshold to define the boundaries of normality in the data space and test whether a test sample comes from the same distribution or not. Gaussian mixed model (GMM) which assumes the objects following a mixture of Gaussian distribution have proven popular [3]. The samples with low probabilities than a specific threshold can be regarded as novelties. Unfortunately, in many real-life scenario, no a priori knowledge of the data distributions is available, Assuming a distribution for training data may be problematic, resulting a poor novelty detection result.

The nearest-neighbour based approach assumes that normal data lie near their neighbourhoods, while potential novelties lie far away from their neighbours. It is a very simple and effective method but the drawback is that it needs to store all the training data points which are further used to compute the distance between a new unseen data point and all the given data points. Angiulli [4] introduced a nearest neighbours based novelty detector. It accepts data points on the basis of their nearest neighbour distances in a training dataset. Tziakos et al. [5] employed the Mahalanobis distance to train a novelty detector and defined a metric to score each vector in a video sequence. Then, the frames in the sequence that score above the threshold were labelled as abnormal.

Another approach called domain-based approach of novelty detection is to find bounded region that contains (almost) all known normal data. A sample is regarded as novelty when it falls outside of the region. Support vector data description (SVDD) which is inspired by the support vector classifier and proposed by Tax and Duin [6], seems to give a flexible and tight data description among the boundary approaches and uses a hypersphere to enclose all objects in one target class with a minimal volume by minimizing the structural risk. A novelty is assessed by determining if a test point lies within the hypersphere. A drawback of these methods is the complexity associated with the computation of the kernel function.

It is noticed that an ensemble of classifiers can actually provide a kind of metric to measure the proximity of a sample and a specific class. As far as we know, there are few works on novelty detection using an ensemble approach. The one of the most outstanding research, using random forest for novelty detection, is proposed by Zhou et al [7]. Zhou et al (2015) made full use of the vote distribution from trees and find a metric to measure the proximity of different samples. Our work is related to this recent work. We make an ensemble system by building n different types of learners and use the class probability vectors from these learners to obtain the mean confidence distribution of each class for novelty detection. Firstly, our method do not rely on the properties of the distribution of data in the training set Furthermore, Our method do not suffer computational complexity like domain-based approaches. Finally, our approach is appropriate to deal with the high-dimensional data. It should be pointed out that Zhou et al [7] use a vlaue generated from vote information of trees to characterize a class while we use a vector generated from probability information from individual learners to charaterize a class.

3 Ensemble learning and confidence distribution

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone [8].

3.1 Confidence distribution

For a classification task, the classifier aims to predict a label from class label set $V_c = \{c_1, c_2, ..., c_k\}$ for a sample *x*. In most cases, an ensemble method constructs a set of base classifiers from the training data and performs classification by taking a vote on the predictions made by base classifiers [7] .

We hypothesis that the ensemble includes T base classifiers $h = \{h_1, h_2, ..., h_T\}$. A k dimensional vector $V_k = \{h_i^1(x); h_i^2(x); ...; h_i^k(x)\}$ are used to represent the predicted outputs of h_i on sample x, where $h_i^j(x)$ is the output of hi on Class cj. Formally,

(i) Majority voting

$$H(x) = c_{\underset{j}{\operatorname{argmax}} \sum_{i=1}^{T} h_i^j(x)} \tag{1}$$

(ii) Weighted voting

$$H(x) = c_{\underset{j}{\operatorname{argmax}} \sum_{i=1}^{T} w_i h_i^j(x)} \quad \text{with} \quad w_i \geq 0, \sum_{i=1}^{T} w_i = 1 \tag{2}$$

where wi is the weigh of hi. A test sample x is classified by taking a majority vote on the individual predictions or by weighting each prediction with the accuracy of the base classifier.

Eq.(1) and Eq.(2) impose no restriction on the output types of hi. In Realistic task, different types of individual learners output different types values of $h_i^j(x)$. There are two commons :

Class label: $h_i^j(x) \in \{0,1\}$, if the hi predict the sample x as class cj, $h_i^j(x) = 1$. Otherwise, $h_i^j(x) = 0$. The voting using class label is called as “hard voting”.

Class probability: $h_i^j(x) \in [0,1]$, corresponds to a estimation of the prior probability $p(c_j | x)$. The voting using class probability is called as “soft voting”.

In an ensemble-based system, it usually assigns a confidence to the decision made by the system. The confidence can be obtained by integrating the outcome of each classifier. which is defined as follows:

$$\text{conf}(C = c_j) = \frac{\sum_{i=1}^{T} h_i^j(x)}{T} \tag{3}$$

where $conf(C = c_j)$ is the confidence of the prediction as Class c_j. *T* is the total number of base classifiers.

Confidence is used to estimate the reliability of predicting a class label for an observation -- the greater the confidence is, the greater the probability of corresponding sample belonging to a class is. A confidence vector for the instance x can be represented as Eq. (4), which represents the confidence distribution generated by an ensemble system.

$$CV_1 = [conf(C = c_1), conf(C = c_2), ..., conf(C = c_k)] \tag{4}$$

4 Ensemble probability distribution novelty detection approach

Based on the aforementioned discussion, it is noted that an ensemble of classifiers is able to provide a kind of metric to measure the proximity between one new sample and known classes. Because samples from the same class have similar confidence distribution, then for one certain class, it have similar confidence distribution. As a result, it can be characterized

by the average or mean confidence distribution of those instances belonging to the same class for a certain class. The proximity between one new sample and a known class can be obtained, based on the distance between sample confidence distribution and the mean confidence of a specific class. A distance threshold need to be preset. While the distance value exceeds the threshold, the sample is rejected by this class. The sample will be regarded as novelty when it is rejected by all known classes. A concrete approach to novelty based on the class probability vectors from component classifiers, denoted as *Ensemble Probability Distribution Novelty Detection* (EPDND). The EPDND algorithm is described in **Algorithm 1.**

Algorithm 1: EPDND algorithm

Require: A training set $D_1 = \{(x_1, y_1),(x_2, y_2),...,(x_m, y_m)\}$ containing k classes with size M_1, M_2...M_k respectively, a testing dataset $D_2 = \{x_1^*, x_2^*,..., x_n^*\}$ with size M_s and learning algorithms: $l_1, l_2,...l_T$

Ensure: A result vector that is used to indicate whether test samples are nomal or not

1 Train T classifiers $h_1, h_2,..., h_T$ on D_1

2 Compute confidence distribution for each class

for t = 1:T

P_t=h_t(D_1) % output the class probability vectors

end for

for i = 1:k

for j = 1:M_i

$$prod_{ij} = \left[p_1^{1j}, p_1^{2j},..., p_1^{kj}, p_2^{1j}, p_2^{2j},..., p_2^{kj},..., p_T^{1j}, p_T^{2j},..., p_T^{kj} \right]$$

end for

$$mprod_i = \left[\sum_{j=1}^{M_i} \sum_{t=1}^{T} p_t^{1j}/(T*M_i), \sum_{j=1}^{M_i} \sum_{t=1}^{T} p_t^{2j}/(T*M_i),..., \sum_{j=1}^{M_i} \sum_{t=1}^{T} p_t^{kj}/(T*M_i) \right]$$

end for

$mprod = [mprod_1; mprod_2;...; mprod_k]$ % confidence distribution matrix

3 Detect novelty according to the *pm*

for $d = 1 : M_s$

$pro^d = [conf_1^d, conf_2^d,..., conf_k^d]$ %sample confidence distribution

$dis^d = distance(pro^d, mprod)$

$dis^d = [dis_1^d, dis_2^d,..., dis_k^d]$

for $i = 1 : k$

if $dis_i^d > t$: **then** d -th sample $\notin$ ith class,denoted by $I_i^d = -1$;

$dis_i^d \le t$: **then** d -th sample $\in$ ith class,denoted by $I_i^d = 1$;

end for

if $\sum_{i=1}^{k} I_i^d = -k$: $result[d] = -1$ % d -th sample $\in$ novelty

else $result[d] = m$ % d -th sample $\in$ *the m -th* where $m = \arg\max_{i=1,2,...,k} \{pm_i^d\}$;

end for

4 Return Result vector *result*

According the **Algorithm 1** above, an ensemble framework are constructed by build n

different individual learners, using training dataset. Then input all training dataset into the n classifiers to get the class probability vectors. For i_{th} class, we take a sum of probability values of each class respectively and the summation are averaged to obtain $mprod_i$ which reflects the mean confidence distribution for i_{th} class. The *mprod* represents the confidence distribution matrix of all classes. For a test sample, the distance between its confidence distribution vector and the $mprod_i$ which is used to measure the proximity of a new sample and a known class is compared with a threshold t to determine whether it belongs to i_{th} class class or not. The distance can be Euclidean distance, cosine similarity or some else. In this paper, we choose the Euclidean distance. If the new instance is rejected by all known classes, it will be predicted as a novelty.

5 Experiment and analysis

5.1 Preliminaries

In this section we present our experimental evaluation of the EPDND framework. In our experiment, we consider five algorithms – Neural Networks (NN), Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM) and Discriminant Analysis Classifier(DAC). And we denote this instance of the framework by EPDND-5.

In order to validate the effectiveness of our proposed methods, we evaluate EPDND-5 by comparing them with the two traditional traditional methods SVDD and GMM. For SVDD, we utilize the tool package libsvm (http://www.csie.ntu.edu.tw/~cjlin/libsvm) developed by Lin Chih-Jen and select RBF as kernel function. For GMM, we set the number of Guassian models as the number of known classes in training dataset, and in initialization, conduct clustering with k-means to determine the GMM components. SVDD, GMM and EPDND-5 are executed on matlab R2015b. We conduct a series of experiments on 4 real-world datasets from UCI. The set of experiments are to demonstrate the effectiveness of the framework on a wide range of real-world datasets. The overall performance of detection could be estimated by *recall*, *accuracy* and F_1 which are defined as Eq. (5), (6).

$$recall = \frac{N_{ia}}{N_{new}} \tag{5}$$

$$accuracy = \frac{N_{ia}}{N_i} \tag{6}$$

$$F_1 = \frac{2 * recall * accuracy}{recall + accuracy} \tag{7}$$

where the N_{new} is the number of total novelties, N_i is the number of samples recognized as novelties and N_{ia} is the number of true novelties among N_i. In order to reflect the overall performance more clearly, we introduce F_1, which is defined as Eq.(7).

5.2 Datasets

UCI datasets: 4 datasets from UCI Data Repository [9] are selected. Some general information about these datasets is shown in Table 1 which lists the details of the UCI datasets.

Table 1. Description of the UCI Real-World datasets and Minist dataset.

Datasets	*#Instances*	*#Attributes*	*#Categories*	*#Class proportion*
Zoo	101	17	7	41/20/5/13/4/8/10
Wine	178	13	3	59/71/48
Balance	625	4	3	49/288/288
Segment	2310	18	7	330/330/330/330/330/330/330

5.3 Experiments

For Wine dataset with 3 classes, as shown in Table 2, we select Class1 as the novelty. For Balance dataset, we randomly select two classes from these 3 classes, and regard each of these two classes as the novelty in turn. For Zoo and Segment dataset, we randomly select 3 classes from 7 classes, and regard each of these 3 classes as the novelty, in turn. The training dataset is also constituted by 70% samples of the known classes, then the remaining 30% of the known classes and part of samples in the novelty class as the testing set. The experiment results on the four UCI datasets are illustrated in Table 2 to Table 5. The best results are denoted in bold.

Table 2. The result of three approach on Wine dataset.

Novelty \ Methods		SVDD	GMM	EPDND
Class1	*recall*	**1**	**1**	0.94
	accuracy	0.71	0.75	0.80
	F_1	0.83	0.86	**0.87**

Table 3. The result of three approach on Balance datasets.

Novelty \ Methods		SVDD	GMM	EPDND
Class1	*recall*	**1**	0.99	**1**
	accuracy	0.76	0.83	0.74
	F_1	0.87	0.90	0.85
Class2	*recall*	0.94	**1**	**1**
	accuracy	0.22	0.22	**0.58**
	F1	0.35	0.36	**0.74**

Table 4. The result of three approach on Zoo datasets.

Novelty \ Methods		SVDD	GMM	EPDND
Class1	*recall*	**1**	**1**	**1**
	accuracy	0.63	0.91	**0.95**
	F_1	0.77	0.95	**0.98**
Class2	*recall*	**1**	**1**	**1**
	accuracy	0.33	0.67	**0.89**
	F1	0.50	0.80	**0.94**
Class3	*recall*	0.80	**1**	**1**
	accuracy	0.89	0.71	**1**
	F1	0.84	0.83	**1**

Table 5. The result of three approach on Segment datasets.

Novelty \ Methods		SVDD	GMM	EPDND
Class1	*recall*	0.84	**1**	0.99
	accuracy	0.92	0.92	**0.97**
	F_1	0.88	0.96	**0,98**
Class2	*recall*	**1**	**1**	0.93
	accuracy	0.93	**0.96**	0.86
	F1	0.97	**0.98**	0.89
Class3	*recall*	0.74	**0.90**	0.83
	accuracy	0.56	0.51	**0.87**
	F1	0.64	0.65	**0.85**

5.4 The result analysis

Experimental results are shown in the Table 2 to 5, which reveals there is no single approach performs best on all datasets. Even on one dataset, for different classes as novelty, no approach outperforms others. In summary, from tables, the best results often achieve when using EPND, although there are some cases that EPND is on equal terms with GMM, and optimal results (the bold in the tables) are rarely achieved by SVDD.

The experimental results on real-world datasets are illustrated in Table 2 to Table 5. For Wine dataset, the EPDND behaves best when the Class1 is regarded as novelties. For Balance dataset, GMM shows prominent advantages while the Class1 is regarded as novelties. However, EPDND has visible advantages to GMM and SVDD in term of F_1 scores while the Class1 is used as new class, though it is not always the best detector. For Zoo, EPDND stands out in terms of 3 performance indexes. Especially when the Class3 of Zoo is used as novelty class, the results are pleased with 3 indexes equalling to 1. For Segment, EPDND achieves the best result in terms of F_1 scores while the Class1 and Class2 are regarded as novelties. According to the results analysis above, our approach outperforms the others in most cases.

6 Conclusions

In this paper, we proposed an efficient framework (EPDND) based on ensemble learning for novelty detection. In particular, we present a specialization of EPND involving 5 different individual classifiers, called EPDND-5, for novelty detection. The probability information from those classifiers are employed to obtain the mean confidence distribution for every class which are used to judge whether a new sample is a novelty.

Extensive experiments show that EPDND achieves superior performance on the novelty detection task. Moreover, EPDND outperforms, in many cases, two commonly used novelty detection approaches, Support Vector Domain Description and Gaussian Mixed Models, in terms of *accuracy*, *recall* and F_1 scores.

There are several avenues for future research. First of all, boosting, Random Forest and bagging can also be considered in our method. Secondly, other proximity measures will be investigated in our future work. Thirdly, since no approach beats its counterparts, meaning no one approach is appropriate to all datasets. The selection of datasets is significant and more datasets should be considered in our experiment to find what kind of method is applicable to what kind of data.

References

1. Pimentel, M. A. F., Clifton, D. A., Lei, C., & Tarassenko, L. (2014) A review of novelty detection, Signal Processing vol **99** pp 215-249.
2. Ding, X., Li, Y., Belatreche, A., & Maguire, L. P. (2014) An experimental evaluation of novelty detection methods, Neurocomputing vol **135** pp 313-327.
3. Lauer, M. (2001) A Mixture Approach to Novelty Detection Using Training Data with Outliers, Machine Learning: ECML pp 300-311
4. Angiulli, F. (2012) Prototype-based domain description for one-class classification IEEE Transactions on Pattern Analysis & Machine Intelligence vol **34** pp 1131-44.
5. Tziakos, I., Cavallaro, A., & Xu, L. Q. (2010) Event monitoring via local motion abnormality detection in non-linear subspace Elsevier Science Publishers B. V.
6. Tax, M. J., Duin, P. W. (1999) Support Vector Domain Description Pattern Recognition Letters vol **20** pp 1191-1199.
7. Zhou, Q. F., Zhou, H., Ning, Y. P., Yang, F., & Li, T. (2015) Two approaches for novelty detection using random forest Expert Systems with Applications vol **42** pp 4840-50.
8. Polikar, & Robi. (2006) Ensemble based systems in decision making IEEE Circuits & Systems Magazine vol **6** pp 21-45.
9. Blake, C. (1998) Uci repository of machine learning databases Neural Information Processing Systems.