Applied Soft Computing 35 (2015) 267–279

Contents lists available at ScienceDirect

Applied Soft Computing

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

Rule-based OneClass-DS learning algorithm

Dat T. Nguyen [a,b,*], Krzysztof J. Cios [a,c]

[a] Department of Computer Science, School of Engineering, Virginia Commonwealth University, Richmond, VA, USA
[b] School of Science & Technology, Hoa-Sen University, Ho Chi Minh City, Viet Nam
[c] IITiS Polish Academy of Sciences, Gliwice, Poland

ARTICLE INFO

Article history:
Received 11 November 2014
Received in revised form 8 April 2015
Accepted 14 May 2015
Available online 3 July 2015

Keywords:
One-class learning algorithm: OneClass-DS
Outlier detection
Anomaly detection
Novelty detection

ABSTRACT

One-class learning algorithms are used in situations when training data are available only for one class, called target class. Data for other class(es), called outliers, are not available. One-class learning algorithms are used for detecting outliers, or novelty, in the data. The common approach in one-class learning is to use density estimation techniques or adapt standard classification algorithms to define a decision boundary that encompasses only the target data. In this paper, we introduce OneClass-DS learning algorithm that combines rule-based classification with greedy search algorithm based on density of features. Its performance is tested on 25 data sets and compared with eight other one-class algorithms; the results show that it performs on par with those algorithms.

Published by Elsevier B.V.

1. Introduction

In a classical supervised learning problem, data instances that represent several classes are available for designing a classifier and a learning algorithm uses this information to discriminate between the classes. In a single-instance learning scenario, the simplest, we have class label for each instance, while in multiple-instance learning scenario we have class label for a group of instances, called bags [1,2]. In the third, more difficult, learning scenario we have data available only for a single class, called Target (each instance belongs to Target). Once a classifier is designed it can be used for predicting whether a new instance belongs to the Target class or not. If an instance does not belong to a Target class it is called an outlier, also known as "ignorance" [3]. Therefore this type of learning problem is known as outlier, or novelty, or anomaly detection. One-class learning algorithms are often designed using information about distribution of the Target training data. Several methods have been developed to solve this challenging problem and they can be divided into density estimation, boundary, reconstruction, and rule-based methods. They differ in their ability to cope with, or exploit, different characteristics of data such as feature scaling, grouping of data into clusters, or taking advantage of data distribution convexity. These methods are briefly described next.

1.1. Density estimation methods

The common and most straightforward method is to estimate density of the training data [4] and then use some threshold to encompass the data. When data are sufficiently large and a density model, such as Parzen, is used this approach works quite well. The drawback, however, is that it requires a large number of instances to overcome the curse of dimensionality [5]. If the dimensionality of data and complexity of the density model are restricted then a large bias is introduced resulting in the model that does not fit the data well. Finding the right model to describe Target data distribution is a bias-variance dilemma. Using the density approach one needs to assume a distribution type, such as Gaussian [6–8], or mixture of Gaussians [5,6,8], or Parzen [9–11]. Algorithm introduced in [1] builds density function from a chosen distribution and then combines this function with a class probability to form an adjusted estimate of the density function of the Target class, which is then used for constructing a decision tree.

1.2. Boundary methods

Boundary methods define a closed boundary around the target data first and then it is optimized. These methods rely heavily on distances between instances and are sensitive to feature scaling [8,12]. Although the volume of data is not always minimized, most of these methods have a strong bias toward a minimal-volume solution. How small the volume is to be depends on the model. The advantage of the boundary methods is that the number of instances required for training is smaller than one required by the density

* Corresponding author.
E-mail addresses: dat.nguyentien1780@hoasen.edu.vn (D.T. Nguyen), kcios@vcu.edu (K.J. Cios).

http://dx.doi.org/10.1016/j.asoc.2015.05.043
1568-4946/Published by Elsevier B.V.

methods. The difficulty, however, is shifted into defining appropriate distance measures. One such example method is the *k*-center, which covers the data with *k* small balls with equal radii [13]. The ball centers, μ_k, are placed on training instances such that the maximum of all minimum distances between training instances and the centers is minimized. To fit the model the following error is minimized:

$$\varepsilon_{k\text{-centre}} = \max_i \left(\min_k \left\| x_i - \mu_k \right\|^2 \right)$$

It uses a greedy search strategy, starting with random initialization. The radius is determined by maximum distance to the instances that the corresponding ball captures. Because of this, the method is sensitive to outliers possibly present in the target data but works well when data have good (compact) clustering structure. However, the user needs to specify in advance both the number of balls, *k*, and the maximum number of tries (the number of runs with random initialization).

Another such method is the nearest-neighbor, NN-d, which is designed from local density estimation by the nearest neighbor classifier [5]. It avoids explicit density estimation and uses only distances to the first nearest neighbors. It is similar to the methods of [14] and [15] that were used for outlier detection in large databases. In the nearest neighbor density estimation a cell, often a hypersphere in a d dimensional space, is centered around the training instance *z*. The volume of this cell is grown until it captures *k* instances of the training data. Support vector machine with RBF kernel was also used for anomaly detection [16]. For each new instance, the method determines if it falls within the learned region: if it does then the instance is considered to be a target instance, otherwise it is an outlier instance. Similar approach, called Robust SVM, was used for intrusion detection [17].

1.3. Reconstruction methods

Reconstruction methods use prior knowledge about data and make assumptions about the data-generating process to build a classifier. The assumption is that a compact representation of the target data can be obtained that decreases noise influence [6,18]. The reconstruction methods assume that outlier instances do not satisfy assumptions about the target class distribution. During testing a new instance may have low or large noise component and thus the corresponding low or high reconstruction error, calculated as a distance to the target training data. Users need to choose appropriate thresholds when using these methods. For example, in learning vector quantization [18] and in *k*-means clustering [6] it is the number of clusters. In self-organizing feature maps [19] it is the dimensionality of the manifold, the number of prototypes per manifold, and the learning rate. In Principal Component Analysis (PCA) [6] it is the mean and basis vectors for each of the subspaces and the noise variance outside of the subspaces. In diabolo networks [20,21] and auto-encoder networks [22] it is the number of layers and neurons, learning rates, and stopping criteria.

1.4. Rule-based methods

Rule-based methods are of key interest here as the introduced below algorithm is rule-based. Rule-based algorithms generate rules that capture target behavior of a system [23–25]. An instance that is not covered by any of the rules is considered an outlier. Different techniques generate rules in different ways. Classification techniques such as IREP and RIPPER [26] learn rules from data by adding outlier class data into the training data. Outlier class is artificially generated so a binary classifier (such as RIPPER) can learn the boundaries between the two classes. In [25,27] supervised outlier detection method was introduced to detect network intrusions while [28] adapted C4.5 algorithm for outlier detection. A similar approach was used in [29] where a query by Bagging method was used.

A similar approach is to use association rule mining algorithm for rule generation, which requires users to specify minimum support and confidence thresholds [30–32]. The advantage of this approach is that it utilizes the fact that outliers occur very rarely in the data, and can be dealt with by choosing appropriate support threshold to ensure that outliers are not taken into account in the process of rule generation. To ensure that rules correspond to strong patterns the rules with low support values are pruned. An application of this technique for intrusion detection was used in ADAM system [30,31] and in [33] for intrusion detection embedded on the network interface card. LERAD method [34] generates association rules from data in the form $P(\text{notW}|\text{U})$, which is conditional probability of one subset of attributes taking on a particular set of values (denoted by notW) given that a disjoint subset of attributes takes on a particular set of values (denoted by *U*). In order to deal with possibly high number of rules that can be generated the authors used sampling and randomization techniques. Similar approach was used for credit card fraud detection and for fraud detection in spacecraft house-keeping data [35]. Observing that an outlying transaction occurs in fewer frequent itemsets when compared with a normal transaction, in [36,37] an approach similar to an outlier detection algorithm for categorical data sets was proposed. The authors also introduced a measure called Frequent Pattern Outlier Factor that ranked the transactions based on the number of frequent itemsets they occur in. A different approach, based on fuzzy classifiers [38–40], uses a fuzzy classifier learning algorithm to create rules from data with noise. For instance, fuzzy classifiers eClass and FLEXFIS-Class [41] can be used with different model architectures for solving one-class problems.

2. OneClass-DS algorithm

In a classical binary classification problem there are positive and negative instances, representing two classes, which are available for training. When rules for the positive (or negative) class are generated they are found based on comparisons between positive and negative instances. During testing, an instance is assigned to either the positive or the negative class. This simple scheme, however, cannot be used for solving one-class classification problems since only instances representing one Target class are available. Moreover, Target instances may contain noise so they can be either true targets or false outliers (FO); this is illustrated in the left rectangle in Fig. 1. Similarly, the Outlier instances (to be identified only during testing) can be either true outliers or false targets; see the right rectangle in Fig. 1.

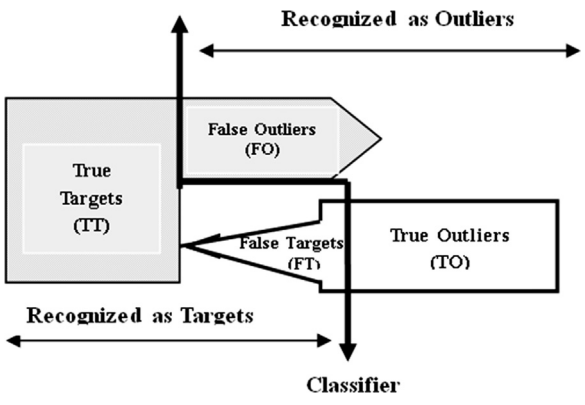

Fig. 1. One-class training and test scenarios.