Clustering Techniques for Outlier Detection

Frank Klawonn
University of Applied Sciences Braunschweig/Wolfenbuettel, Germany

Frank Rehm
German Aerospace Center, Germany

INTRODUCTION

For many applications in knowledge discovery in databases, finding outliers, which are rare events, is of importance. Outliers are observations that deviate significantly from the rest of the data, so they seem to have been generated by another process (Hawkins, 1980). Such outlier objects often contain information about an untypical behaviour of the system.

However, outliers bias the results of many data-mining methods such as the mean value, the standard deviation, or the positions of the prototypes of *k-means* clustering (Estivill-Castro & Yang, 2004; Keller, 2000). Therefore, before further analysis or processing of data is carried out with more sophisticated data-mining techniques, identifying outliers is a crucial step. Usually, data objects are considered as outliers when they occur in a region of extremely low data density.

Many clustering techniques that deal with noisy data and can identify outliers, such as possibilistic clustering (PCM) (Krishnapuram & Keller, 1993, 1996) and noise clustering (NC) (Dave, 1991; Dave & Krishnapuram, 1997), need good initializations or suffer from lack of adaptability to different cluster sizes. Distance-based approaches (Knorr & Ng, 1998; Knorr, Ng, & Tucakov, 2000) have a global view on the data set. These algorithms can hardly treat data sets that contain regions with different data density (Breuning, Kriegel, Ng, & Sander, 2000).

In this work, we present an approach that combines a fuzzy clustering algorithm (Höppner, Klawonn, Kruse, & Runkler, 1999) or any other prototype-based clustering algorithm with statistical distribution-based outlier detection.

BACKGROUND

Prototype-based clustering algorithms approximate a feature space by means of an appropriate number of prototype vectors, where each vector is located in the centre of the group of data *(the cluster)* that belongs to the respective prototype. Clustering usually aims at partitioning a data set into groups or clusters of data, where data assigned to the same cluster are similar and data from different clusters are dissimilar. With this partitioning concept in mind, an important aspect of typical applications of cluster analysis is the identification of the number of clusters in a data set. However, when we are interested in identifying outliers, the exact number of clusters is irrelevant (Georgieva & Klawonn). The idea of whether one prototype covers two or more data clusters or whether two or more prototypes compete for the same data cluster is not important as long as the actual outliers are identified and assigned to a proper cluster. The number of prototypes used for clustering depends, of course, on the number of expected clusters but also on the distance measure respectively the shape of the expected clusters. Because this information is usually not available, the Euclidean distance measure is often recommended with rather copious prototypes.

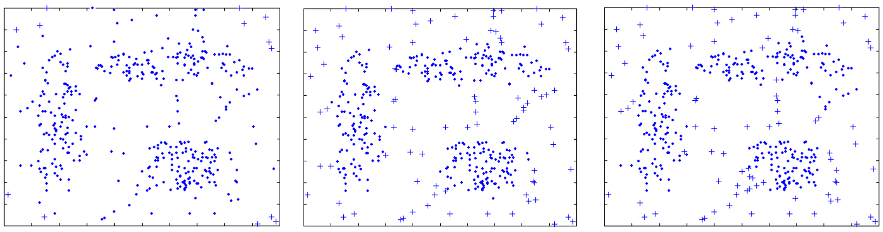
One of the most referred statistical tests for outlier detection is the Grubbs' test (Grubbs, 1969). This test is used to detect outliers in a univariate data set. Grubbs' test detects one outlier at a time. This outlier is removed from the data set, and the test is iterated until no outliers are detected.

MAIN THRUST

The detection of outliers that we propose in this work is a modified version of the one proposed in Santos-Pereira and Pires (2002) and is composed of two different techniques. In the first step we partition the data set with the fuzzy *c*-means clustering algorithm so the feature space is approximated with an adequate number of prototypes. The prototypes will be placed in the centre of regions with a high density of feature vectors. Because outliers are far away from the typical data, they influence the placing of the prototypes.

After partitioning the data, only the feature vectors belonging to each single cluster are considered for the detection of outliers. For each attribute of the feature vectors of the considered cluster, the mean value and the standard deviation has to be calculated. For the vector

Figure 1. Outlier detection with different numbers of prototypes



(a) 1 prototype (b) 3 prototypes (c) 10 prototypes

with the largest distance[1] to the mean vector, which is assumed to be an outlier, the value of the z-transformation for each of its components is compared to a critical value. If one of these values is higher than the respective critical value, then this vector is declared an outlier. One can use the Mahalanobis distance as in Santos-Pereira and Pires (2002), but because simple clustering techniques such as the fuzzy c-means algorithm tend to spherical clusters, we apply a modified version of Grubbs' test, not assuming correlated attributes within a cluster.

The critical value is a parameter that must be set for each attribute depending on the specific definition of an outlier. One typical criterion can be the maximum number of outliers with respect to the amount of data (Klawonn, 2004). Eventually, large critical values lead to smaller numbers of outliers, and small critical values lead to very compact clusters. Note that the critical value is set for each attribute separately. This leads to an axes-parallel view of the data, which in cases of axes-parallel clusters leads to a better outlier detection than the (hyper)spherical view of the data.

If an outlier is found, the feature vector has to be removed from the data set. With the new data set, the mean value and the standard deviation have to be calculated again for each attribute. With the vector that has the largest distance to the new centre vector, the outlier test will be repeated by checking the critical values. This procedure will be repeated until no outlier is found. The other clusters are treated in the same way.

Figure 1 shows the results of the proposed algorithm. The crosses in this figure are feature vectors, which are recognized as outliers. As expected, only a few points are declared as outliers when approximating the feature space with only one prototype. The prototype will be placed in the centre of all feature vectors. Hence, only points on the edges are defined as outliers. Comparing the solutions with 3 and 10 prototypes, you can determine that both solutions are almost identical. Even in the border regions, were two prototypes competing for some data points, the algorithm would rarely identify these points as outliers, which they intuitively are not.

FUTURE TRENDS

Figure 1 shows that the algorithm can identify outliers in multivariate data in a stable way. With only a few parameters, the solution can be adapted to different requirements concerning the specific definition of an outlier. With the choice of the number of prototypes, it is possible to influence the result in a way that with lots of prototypes, even smaller data groups can be found. To avoid overfitting the data, it makes sense in certain cases to eliminate very small clusters. However, finding out the proper number of prototypes should be of interest for further investigations.

In the case of using a fuzzy clustering algorithm such as FCM (Bezdek, 1981) to partition the data, it is possible to assign a feature vector to different prototype vectors. In that way, you can consolidate whether a certain feature vector is an outlier if the algorithm decides for each single cluster that the corresponding feature vector is an outlier.

FCM provides membership degrees for each feature vector to every cluster. One approach could be to assign a feature vector to the corresponding clusters with the two highest membership degrees. The feature vector is considered as an outlier if the algorithm makes the same decision in both clusters. In cases where the algorithm gives no definite answers, the feature vector can be labeled and processed by further analysis.

CONCLUSION

In this article, we describe a method to detect outliers in multivariate data. Because information about the number and shape of clusters is often not known in advance, it is necessary to have a method that is relatively robust with respect to these parameters. To obtain a stable algorithm, we combined approved clustering techniques, including the FCM or k-means, with a statistical method to detect outliers. Because the complexity of the presented algorithm is linear in the number of points, it can be applied to large data sets.