Outlier Detection Integrating Semantic Knowledge

Zengyou He, Shengchun Deng, and Xiaofei Xu

Department of Computer Science and Engineering, Harbin Institute of Technology,
Harbin 150001, P. R. China
{zengyouhe@yahoo.com, dsc@hope.hit.edu.cn,
xiaofei@hope.hit.edu.cn}

Abstract. Existing proposals on outlier detection didn't take the semantic knowledge of the dataset into consideration. They only tried to find outliers from dataset itself, which prevents finding more meaningful outliers. In this paper, we consider the problem of outlier detection integrating semantic knowledge. We introduce new definition for outlier: *semantic outlier*. A semantic outlier is a data point, which behaves differently with other data points in the same class. A measure for identifying the degree of each object being an outlier is presented, which is called *semantic outlier factor (SOF)*. An efficient algorithm for mining semantic outliers based on *SOF* is also proposed. Experimental results show that meaningful and interesting outliers can be found with our method.

1 Introduction

An outlier in a dataset is defined informally as an observation that is considerably different from the remainders as if it is generated by a different mechanism. Mining for outliers is an important data mining research with numerous applications, including credit card fraud detection, discovery of criminal activities in electronic commerce, weather prediction, marketing and customer segmentation.

Recently, a few studies have been proposed on outlier detection (e.g. [1], [2], [3], [4]) from the data mining community.

Distance-based outlier is presented in [1]. A distance-based outlier in a dataset D is a data object with *pct%* of the objects in D having a distance of more than d_{min} away from it. This notion generalizes many notions from distribution-based approach and enjoys better computational complexity. In [2], it is further extended based on the distance of a point from its k^{th} nearest neighbor. After ranking points by the distance to its k^{th} nearest neighbor, the *top k* points are identified as outliers. Efficient algorithms for mining *top-k* outliers are given.

In [3], the concept of "*local outlier*" is introduced. The outlier rank of a data object is determined by taking into account the clustering structure in a bounded neighborhood of the object, which is formally defined as "*local outlier factor*" (*LOF*). In [4], the authors consider the problem of outlier detection in subspace to overcome dimensionality curse.

All of these works compute outliers without considering semantic knowledge that can be used to facilitate the mining process. In fact, many datasets contain attributes that can be regarded as the class label of the dataset. For example, the Congressional

X. Meng, J. Su, and Y. Wang (Eds.): WAIM 2002, LNCS 2419, pp. 126-131, 2002.

Voting dataset in the UCI Machine Learning Repository [6] has a classification label of Republican or Democrat provided with each record. Obviously, the records with the same class label should be similar with each other from the semantic knowledge that the people in the same group should have similar ideas. Thus, if we run a clustering algorithm on the voting dataset, it is expected that the records in every output cluster be identified with the same class label. However, it is often not this case. It is reasonable to take those records whose class labels are different from that of the majority of the cluster as outliers. The above observation forms our basic idea.

In this paper, we introduce new definition for outlier: *semantic outlier*. Informally, a semantic outlier is a data point, which behaves differently with other data points in the same class. An effective algorithm for mining semantic outlier is also proposed.

Contributions of this paper are as follows:

- We propose new definition for outlier-*semantic outlier,* which has great new intuitive appeal and numerous applications.
- A measure for identifying the degree of each object being an outlier is presented, which is called *semantic outlier factor (SOF)*.
- We propose an efficient algorithm for mining *semantic outliers* based on our definition.

The remainder of this paper is organized as follows. In Section 2, we formalize our definition of *semantic outlier* in detail. Section 3 presents the algorithm for mining defined outliers. Experimental results are given in Section 4. Section 5 concludes the paper.

2 Formal Definition of Semantic Outlier

Before we define the concept of semantic outliers and design the measure for outlier factor, let's first look at the concept about clustering.

Definition 1 Let $A_1, \ldots, A_m$ be a set of attributes with domains $D_1,\ldots, D_m$ respectively. Let the dataset D be a set of records where each record t: $t \in D_1 \times \ldots \times D_m$. The results of a clustering algorithm executed on D is denoted as: $C = \{C_1, C_2, \ldots, C_k\}$ where $C_i \cap C_j = \phi$ and $C_1 \cup C_2 \cup \ldots, \cup C_k = D$. The number of cluster is k.

Definition 2 Suppose CL is an additional attribute for D, which distinguishes the class of records and has the set of different attribute values $\{cl_1, cl_2, \ldots, cl_p\}$. The output $C = \{C_1, C_2, \ldots, C_k\}$, just as described in Definition 1 will be produced if a clustering algorithm is executed on D. We define $Pr(cl_i|D)$ and $Pr(cl_i|C_j)$ as the frequency of cl_i in D and frequency of cl_i in C_j.

$$Pr(cl_i|D) = \frac{|\{t \mid t.CL = cl_i, t \in D\}|}{|D|} \quad (1)$$

$$Pr(cl_i|C_j) = \frac{|\{t \mid t.CL = cl_i, t \in C_j\}|}{|C_j|} \quad (2)$$

As we have argued in Section 1, if we run a clustering algorithm on the dataset D, it is expected that the records in every output cluster should be identified with the same

class label. However, it is often not this case. It is reasonable to take those records whose class labels are different from that of the majority of the cluster as outliers. The value $Pr(cl_i|C_j)$ represents the ratio of those records with class label cl_i in cluster C_j. Therefore, if this value is relative small, it indicates that records with label cl_i are likely to be outliers.

Definition 3 Given a set of records R and a record t, the *similarity* between R and t is defined as:

$$sim(t,R) = \frac{\sum_{i=1}^{|R|} similarity(t,T_i)}{|R|} \quad \text{where } \forall T_i \in R. \quad \textbf{(3)}$$

In this paper, we take the *average similarity* of a record with each record in a set of records as the similarity between the specified record and the set. As to the similarity between records, the measure used in the clustering algorithm can be adopted.

Definition 4 (*semantic outlier factor of a record t*) supposes the clustering algorithm assign t to C_k and the class value of t is cl_i. And R is the subset of D with class value cl_i. The *semantic outlier factor of a record t* is defined as:

$$SOF(t) = \frac{\Pr(cl_i \mid C_k) * sim(t,R)}{\Pr(cl_i \mid D)} \quad \textbf{(4)}$$

The meanings of $Pr(cl_i|D)$, $Pr(cl_i|C_j)$ and $sim(t,R)$ are the same as described in Definition 2 and Definition 3. We will give the rationale of defining outlier factor as (4) as follows.

As we have shown, the value $Pr(cl_i|C_k)$ represents the ratio of those records with class label cl_i in cluster C_j. Therefore, if this value is relative small, it indicates that records with label cl_i are likely to be outliers.

To avoid the discrimination of class with small size, the $Pr(cl_i|D)$ is used to adjust the balance. For example, suppose $|D|=100$, $Pr(cl_1|D)=0.4$ and $Pr(cl_1|D)=0.6$. If we get a cluster $|C_1|=50$ with $Pr(cl_1|C_1)=0.4$ and $Pr(cl_2|C_1)=0.6$. It is clear that without consideration of the frequencies of class values in the whole dataset, those records with class label cl_1 are more likely to be taken as outliers than those with class label cl_2. However, this is not the case. From the point of view of statistics, they should be given equal chance.

For the definition of *semantic outlier* considers the outlier semantically, the measure $sim(t, R)$ describes how records differ from others in the same class. Obviously, if this value is relative small, it indicates that this record is likely to be an outlier.

Without considering the effect of the choice of clustering algorithm, the value of $Pr(cl_i|D)$ is determined by the characteristics of dataset. For clustering algorithm group similar records into the same cluster, intuitively, we can get the relationship between $Pr(cl_i|C_j)$ and $sim(t,R)$ that: "Smaller $Pr(cl_i|C_j)$ leads to smaller $sim(t,R)$ or the reverse".

According to the definition of *SOF*, it is easy to see that outliers are those records with smaller values of *SOF*.

3 *FindSOF*: The Algorithm for Detecting Semantic Outliers

With the semantic outlier factor *SOF*, we can determine the degree of a record's deviation. In this section, we will describe our algorithm for detecting semantic outliers according to Definition 4.

Algorithm *FindSOF*

Input: $D(A_1, \ldots, A_m, CL)$, // the dataset
CL: the class label // semantic knowledge

Output: *The values of SOF for all Records* //indicates the degree of deviation

/* $\{cl_1, cl_2, \ldots, cl_p\}$: the set of different attribute values for CL */
(cl_i, C_j)_counter: records the number of cl_i in cluster C_j
(cl_i, D)_counter: records the number of cl_i in dataset D

```
01 begin
02      /* Using clustering algorithm to partition the dataset C = {C1, C2, …, Ck}*/
03      foreach record t do begin
04          for the attribute CL i=1 to p do begin
05              if t.CL = cli then
06                  (cli, Cj )_counter++      // t in Cj
07                  (cli, D)_counter++
08                  compute the similarity of sim (t, (cli, D))
09      end
10
11      /* compute SOF (t) */
12      foreach record t do begin      // t.CL = cli
13          SOF = ComputeSOF ((cli, Cj )_counter, (cli, D)_counter, sim (t, (cli, D)))
14          Label on the disk
15      end
16 end
```

Fig. 1. The Algorithm *FindSOF*

To compute *SOF (t)*, we only need to count and compute three values, $Pr(cl_i|C_k)$, $Pr(cl_i|D)$ and *sim (t, R)*. Let us assume that there are $m+1$ attributes, $A_1, \ldots, A_m$ and

CL. Attribute *CL* is the class label. For each value cl_i of *CL*, $1 \le i \le p$, $k+1$ counters are maintained for recording the number of cl_i in $C_1, C_2, \ldots, C_k$ and *D*.

The Algorithm *FindSOF* is listed in Fig.1. It first partitions the dataset into clusters with a clustering algorithm (line 2). Then, it scans the table twice. In the first pass over the dataset, all counters are updated and the similarity of the record with other records in the same class is computed (line 3-9). In the second pass, the value of *SOF* for each record is computed and labeled on the disk (line 12-15). It is clear that, according to definition 4, the value of *SOF* will be computed as:

$$SOF(t) = \left(\frac{(cl_i, C_j)\_counter}{|C_j|}\right) * sim(t, (cl_i, D)) * \left(\frac{|D|}{(cl_i, D)\_counter}\right)$$

Algorithm *FindSOF* has three parts: 1) Clustering the dataset, 2) Updating counters and 3) Computing the values of *SOF*. The selected clustering algorithm determines the cost of part 1. We can assume it to be $O(H)$. As the part 2 and part 3, one scan over the dataset is required separately. Therefore, the runtime for the algorithm is $O(H+N)$, where *N* is number of the records in the dataset.

4 Experimental Results

In this section, we show that our ideas can be used to successfully identify outliers, which appear to be meaningful.

We experimented with real-life dataset: the Congressional Voting dataset, which was obtained from the UCI Machine Learning Repository [6]. Now we will give a brief introduction about this dataset.

It is the United States Congressional Voting Records in 1984. Each record represents one Congressman's votes on 16 issues. All attributes are boolean with Yes (donated as *y*) and No (donated as *n*) values, and missing value is donated as "?". A classification label of Republican or Democrat is provided with each record. The dataset contains 435 records with 168 Republicans and 267 Democrats.

As pointed out in Section 1, the records with the same class label should be similar with each other from the semantic knowledge that the people in the same group should have similar ideas. Thus, the measure *SOF* can be used to detecting outliers.

The clustering algorithm we used here is the *Squeezer* [5] algorithm, which can produce good clustering results for categorical dataset. We ran algorithm *FindSOF* to get top 10 semantic outliers. The strongest outlier among them is the record with *SOF* = 0.32 (republican, y, y, y, y, n, n, y, y, y, y, y, n, n, y, n, y, 2, 0.32). His group reveals that his votes should be similar with other republicans. However, from the value distribution of the dataset and his votes we can see that, only on the 4, 10, 13, 15, 16 issues he behaved like most of the republicans, while on the issues 1,3, 5, 7,8,9, 12,13, he acted as he is a democrat. He is so similar to democrats, not republicans.

Other outliers can be analyzed in the same manner. It shows that the *FindSOF* algorithm can find interesting and meaningful outliers.

In contrast, we implemented the algorithm described in [2] to find outliers according to their definitions. The results show that all the top 10 semantic outliers can't be found with the method in [2], which indicates that previous algorithms failed to find these meaningful and interesting outliers. Thus, we believe that our method is promising.

5 Conclusions

Existing proposals on outlier detection didn't take the semantic knowledge of dataset into consideration. They only tried to find outliers from the dataset itself, which prevents finding more meaningful outliers. In this paper, we consider the problem of outlier detection integrating semantic knowledge. We introduce new definition for outlier: *semantic outlier*. A semantic outlier is a data point, which behaves differently with other data points in the same class. We also give measures for identifying the degree of being outliers- *SOF* and effective algorithm for mining semantic outliers based on it.

However, it is needed to point out that, other categories of outliers discussed in Section 2 are also useful and interesting in their own rights. The semantic outliers proposed here can be regarded as complementary to other kinds of outliers in real applications.

In the future work, we will integrate the *SOF* algorithm with clustering algorithms to make the detecting process more efficient. Effective *top-k* outliers' detection algorithms will be also addressed.

Acknowledgements. The National Nature Science Foundation of China (No. 60084004) and the IBM SUR Research Fund supported this research.

References

1. E. M. Knorr, R. T. Ng: Algorithms for Mining Distance-Based Outliers in Large Datasets. Proc. 24th Int. Conf. on Very Large Database, New York, NY, 1998, pp. 392-403.
2. S. Ramaswamy, R. Rastogi, S. Kyuseok: Efficient Algorithms for Mining Outliers from Large Data Sets. *Proc. ACM SIGMOD 2000 Int. Conf. on Management of Data*, Dallas, Texas, 2000.
3. M. M. Breunig, H. P. Kriegel, R. T. Ng, J. Sander: LOF: Identifying Density-Based Local Outliers". Proc. ACM SIGMOD 2000 Int. Conf. on Management of Data, Dallas, Texas, 2000.
4. C. Aggarwal, P. Yu: Outlier Detection for High Dimensional Data. Proc. of the 2001 ACM SIGMOD Int'1 Conf. Management of Data, pp. 37-46, Santa Barbara, CA, USA.
5. Z. He, S. Deng and X. Xu: *Squeezer:* An Efficient Algorithm for Clustering Categorical Data. Technical Report, HIT, 2001. http://202.118.239.67/tech/squeezer.pdf. To appear in Journal of Computer Science and Technology.
6. C. J. Merz, Murphy: UCI Repository of Machine Learning Databases. (Http://www.ics.uci.edu/~mlearn/MLRRepository.html).