A Robust Method of Detecting DB-Outliers in High Dimensional Datasets

Yuan Li[+] Hiroyuki Kitagawa[+,++]

[+]Graduate School of Systems and Information Engineering
[++]Center for Computational Sciences
University of Tsukuba, Tennoudai 1-1-1, Tsukuba, Ibaraki, 305-8573 Japan

1 Introduction

Outlier detection aims to discover abnormal objects in datasets. Unusual objects often involve useful information that may lead to some serious incidents. Techniques of outlier detection can be used in many popular applications such as credit card fraud, money laundering and so on. Such applications generally process high dimensional datasets. Most traditional research attempts to define algorithms based on distance [2] or density [4]. However, it has been indicated that in high dimensional spaces, data distribution is sparse, which means that every data point becomes a good outlier candidate based on the definition of distance or density. Consequently, traditional algorithms become impractical when processing high dimensional datasets.

It has been certified that meaningful outliers are likely to be defined by examining the behavior of the data in low dimensional projections [3]. Therefore, we employ the Subspace-Based method [3, 6] to address the curse of dimensionality.

On the other hand, in most cases, a few decisive predetermined parameters are necessary for outlier detection algorithms. Such parameters are usually key factors in detecting outliers, so correspondingly they are quite difficult to determine beforehand. The Example-Based method is shown to be promising in discovering hidden user views of outliers [1]. Hence, outlier examples are good usable inputs as a substitute for vital parameters. Further, we try to use outlier examples to mitigate the curse of dimensionality.

We propose a method that first finds an optimal subspace where outlier examples are outstanding more significantly than in any other subspaces. Then it reports objects having similar characteristics to examples in the optimal subspace as outliers. Actually, outlier detection is operated on the optimal subspace with lower dimensionality. Moreover, our method is an improved robust approach that can abide noise outlier examples.

2 Fundamental Concept

The notion of DB-Outlier studied here is the same as Knorr and Ng's work [2]:

An object O in a dataset T is a DB(p, D)-Outlier if at least fraction p of the objects in T lie greater than distance D from O.

The parameter p is the minimum fraction of objects in a dataset that must reside outside an outlier's D-neighborhood. For convenient explanation and calculation, we employ another parameter M, which denotes the maximum portion of objects within an outlier's D-neighborhood. M can be computed as follows:

$$M = N(1-p) \qquad N : datasize \tag{1}$$

The detection of DB-Outliers can be elucidated as detecting those objects that have no more than M neighbors in their D-neighborhoods.

3 Proposed Method

The main inputs of our method are outlier examples. First, we look for the most suitable subspace where outlier examples are isolated more significantly than in any other subspaces. Such an optimal subspace generally has low dimensionality because outlier examples have only a portion of the exceptional attributes in the real world. If noise is interfused in outlier examples, we prefer that the subspace be the optimal subspace, where as many outlier examples perform abnormal appearances as possible. After discovering the optimal subspace, we seek objects that also reside in sparse areas, just as outlier examples in this subspace. Such objects having similar characteristics to outlier examples will be reported as outliers.

There is a problem in looking for the most suitable subspace: before examining all subspace candidates whose dimensionalities vary from 1 to the total dimension, we cannot confirm which subspace is the one required. If the total dimension is very high the brute force method, which checks all candidate combinations, is infeasible. For this reason, we exploit a Genetic Algorithm (GA) [5] to select the optimal subspace in less time.

3.1 Procedure of Proposed Method

This section overviews the procedure of our proposed method. There are three main steps in our proposed method: (1) Detecting Optimal Subspace with a GA, (2) Parameter Selection, (3) Outlier Report.

3.2 Detecting Optimal Subspace with a GA

Here we introduce how to find the optimal subspace. This task is accomplished by a GA, There are three steps in a GA: Selection, Crossover, and Muttation. A Fitness Value function is used in evaluating solutions. It is relevant to screening out preferable solutions. Defining a proper Fitness Value function is a crucial component of the GA. Here, we demonstrate the calculation of Fitness Value.

Let Nm denote the number of D-neighbors. When the value of parameter D grows longer from 0, the number of D-neighbors, namely Nm of an isolated point, has poor growth for a certain time. In contrast, the value of Nm of an ordinary object mushrooms as the D grows from the word go, since the density of a normal object's D-neighborhood is high. According to this property, we build a Distance-Nm chart for each solution subspace. A D-Nm (Distance-Nm) line describes the relationship between distance D and Nm of an object. Isolated objects have different types of D-Nm lines from the normal type. In the optimal subspace, outlier examples are isolated, so their D-Nm lines stay away from those of normal objects.

Figure 1 describes a Distance-Nm space. In this figure, two curved lines stand for D-Nm lines of an outlier example and an ordinary object, respectively. The more the outlier examples are isolated, the farther the two lines keep away from each other. Let A_{NO} be the area encircled by D-Nm lines of outlier examples and normal objects. We use the value of A_{NO} to measure the goodness of outlier examples. That means A_{NO} can also be used to examine subspaces.

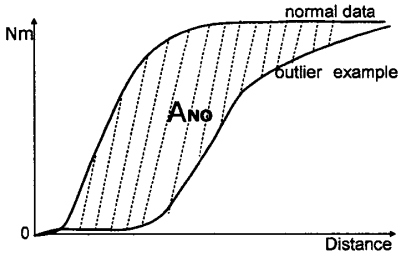

Figure 1 Distance-Nm Lines

If noise is interfused in outlier examples, we cannot find

such a subspace in which all the outlier examples exhibit abnormal behavior. A_{NO} of a noise is extremely small, so it differs significantly from those of isolated examples. Taking noises into account, the optimal subspace should contain as many isolated outlier examples as possible. Further, it should have a large A_{NO} value. Also, users usually wish to find a subspace with lower dimensionality. Consequently, we define the Fitness Value function as follows:

$$f = \frac{A_{NO} * C_o}{k * C_e} \quad (2)$$

f: Fitness Value.
A_{NO}: *Area encircled by D-Nm lines of outlier examples and normal objects.*
C_o: *number of outlier examples isolated in the subspace.*
C_e: *number of all outlier examples.*
k: *dimensionality of a dataset.*

The GA repeat Selection, Crossover and Mutation until it reaches convergence criterion that a certain percentage of solutions' Fitness Values become the same, and the solution which has the largest Fitness Value stands for the optimal subspace.

3.3 Parameter Selection

We still interpret this procedure with a Distance-Nm chart. First, we randomly select points on the distance line between $\frac{max}{2}$ lines (in Figure 2, they are shown as D1 and D2). $\frac{max}{2}$ lines mark the positions where the gaps between outlier examples' and normal objects' D-Nm lines are half of the maximum gap. We then trace out vertical lines with these selected distance points. Next, we also randomly choose one point on each vertical line segment surrounded by D-Nm lines of outlier examples and normal objects. Each point on the vertical line decides a pair of (M, D). Such points can separate outliers having similar characteristics to outlier examples from normal data. Parameter p can be easily computed by function $p = 1 - M/N$. In Figure 2; two points produce two different pairs of (p, D).

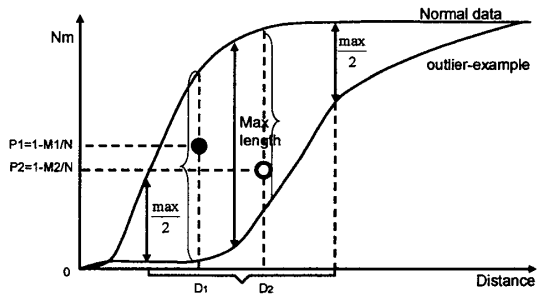

Figure 2 Selection of Parameters (p,D)

3.4 Outlier Report

For the same dataset, different pairs (p, D) bring different results. To make our results more accurate, we detect DB-Outliers with several pairs of (p, D) produced in the "Parameter Selection," and not only report outliers detected in the optimal subspace, but also the "outlierness" degree.

4 Experimental Evaluation

We test our proposed method on both synthetic and real datasets, and also compare our new method with the previous one over some synthetic and real datasets. This section mainly demonstrate a part of experiment results to show our method is effective and efficient over real datasets. Figure 3 shows the results of the abalone dataset.

Table 1 is a summary evaluation of our proposed method. Dim is the dimensionality of the optimal subspace, p_Size stands for population size of the GA and

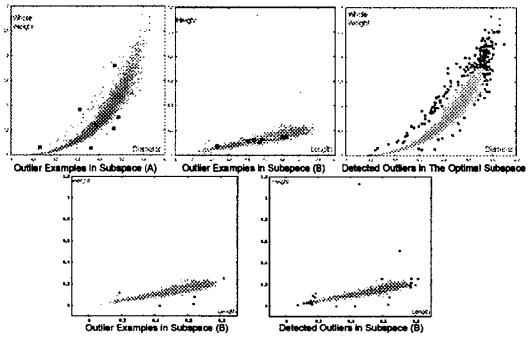

Figure 3 Result of Abalone Dataset

Acc describes the probability of discovering the optimal subspaces during 10 trials. Since there are three suitable subspaces of the NBA data based on user examples, the Accuracy of the NBA dataset denotes the number of times the best three subspaces are found. Sub_T records the average time to discover the optimal subspace.

Table 1 Summary of the Proposed Method

Dataset	Dim	p_Size	Acc	Sub_T(ms)
Synthetic Data	2	80	100%	286547
	3	80	100%	690281
	4	80	100%	759434
Abalone Data	2	100	100%	218599
NBA Data	2	100	100%	80755
Exchange Rate Data	2	50	100%	15635

5 Conclusions and Future Work

In this paper, we discussed a method to detect outliers from high dimensional datasets under users' intentions. Most traditional techniques need decisive parameters to be decided in advance. Actually, such crucial parameters are generally not easily predefined. Thus, we utilize user-provided outlier examples to propose a method whose central ideas are detecting an optimal subspace where these examples show more abnormal behaviors than in others, and picking out outliers having similar characteristics to examples.

Improvement to address the problem of processing time is our future research issue. In the future, we plan to do research on producing hybrid methods to improve processing time.

Acknowledgements

This research has been supported in part by the Grant-in-Aid for Scientific Research from JSPS(#18200005) and MEXT(#19024006).

References

[1] C. Zhu, H. Kitagawa, S. Papadimitriou, and C. Faloutsos. OBE: Outlier By Example. *Proc. PAKDD 2004, LNAI 3056*, pp.222-234, 2004.

[2] E. M. Knorr and R. T. Ng. Algorithms for Mining Distance-Based Outliers in Large Datasets. *Proc. VLDB*, pp.392-403, 1988.

[3] C. C. Aggarwal and P. S. Yu. Outlier Detection for High Dimensional Data. *Proc. SIGMOD Conf.*, pp.37-46, 2001.

[4] M. M. Breuning, H. P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying Density-Based Local Outliers. *Proc. SIGMOD Conf.*, pp.93-104, 2000.

[5] D. E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. *Addison Wesley*, 1989.

[6] C. C. Aggarwal and P. S. Yu. An Effective and Efficient Algorithm for High-dimensional Outlier Detection. *The VLDB Journal*, Vol. 14, No. 2, pp.211-221, 2005.