Ranking Optimization using Outlier Detection Technique

Yogendra Mahate
SOIT, UTD, RGPV
Bhopal, MP, India

Roopam Gupta
SOIT, UIT, RGPV
Bhopal, MP, India

Sanjeev Sharma
SOIT, UTD, RGPV
Bhopal, MP, India

ABSTRACT

Product ranking optimization is an emerging required research area where we are getting a heavy duly competition day by day, number of products with sort of ranges are available in market, finding the best out of the different product is a major problem, several times it observed that number of survey has been made in order to make a product ranking and show it to users about the best available product for their requirement in the market, the ranking been held on multiple required attributes work or play the role to increase or decrease their ranking.

Existing Line-up technique is only provided rank to number of products. It did not optimize the ranking. Now, in this paper, defeat these entire problems and provide the best solution.

Introduce the outlier detection technique to optimize the product ranking according to requirement feature or attribute. Here i am going to implement some algorithm to detect best outliers such as distributing solving set algorithm, line-up algorithm & improved LAZY DSS.

Keywords

Outliers, Ranking optimization, multi attribute ranking, solving set algorithm, line-up algorithm

1. INTRODUCTION

Very frequently, there exist data objects that do not comply with the general behavior or model of the data. Such data objects, which are heavy different from or inconsistent with the continuing set of data, are called outliers. An outlier is a data set which is dissimilar from the remaining data. The outlier is also mentioned to as disfigurement, deviants or anomalies in the data extraction and statistics literature. In most applications the data is produced by one or more getting processes, which will reflect activity in the system or observations collected about entities. When the getting process behaves in an insouciant way, it results in the creation of outliers. Thus, an outlier often contains useful information about anomaly characteristics of the arrangements and entities, which shock the data generation process.

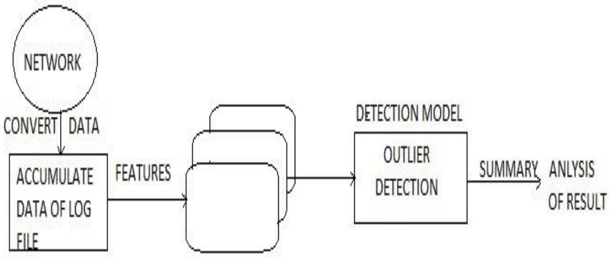

Figure 1. Outlier detection process in Data Mining

The realization of such strange characteristics provides useful application-specific insights.

Some examples of this are as follows:

Intrusion Detection Systems:

In many host-based or networked computer systems, different kinds of data are gathered about the operating system calls, network traffic and other action in the organization. This data may show unusual behavior because of malicious activity.

Credit Card Fraud:

Credit card fraud is quite prevalent, because of the simplicity with which sensitive information such as a credit card number may be compromised. This generally leads to unauthorized use of the credit card. In many cases, unauthorized use will show different patterns, such as a purchasing spree from geographically obscure locations. Such forms can be applied to find outliers in credit card transaction data.

Interesting Sensor Events:

Sensors are often used to track various environmental and location parameters in many real diligence. The sudden alterations in the underlying patterns may represent cases of interest. Case detection is one of the primary motivating applications in the area of sensor webs.

In all these applications, the data has a “normal” model, and anomalies are recognized as deviations from this normal model. In many cases such as an intrusion or fraud detection, the outliers can only be observed as a sequence of multiple data points, sort of than as an individual data point. For example, a fraud event may frequently reflect the actions of an individual in a particular sequence. The specificity of the sequence is relevant to identifying the anomalous effect. Such anomalies are as well mentioned to as collective anomalies, because they can simply be inferred collectively from a set or sequence of data points. Such collective anomalies typically represent unusual events, which need to be observed from the data. The outlier’s distribution method for detecting distance-based outliers in huge data sets. This approach is based on the concept of outlier detection solving set, which is a diminished subset of the data set that can be also employed for predicting fresh outliers. The method exploits parallel computation in order to obtain vast time savings. Indeed, beyond taking up the rightness of the consequence, the proposed schema exhibits excellent executions. Basic disadvantages of the distance-based algorithms are the majority of these algorithms have quadratic complexity and modern information system contain heterogeneous data of complex structure. The biggest cons of distance-based outliers is depends on previous parameters. Now, using kernel functions to overcome to all disadvantages of its. The function of kernel functions allows us to avoid calculating and storing images in the explicit form, which reduces computational resources & storage required. The fact of kernel function is not fixed makes it possible to adapt various data mining algorithms is based on the calculation of inner product or distance for a wide set of problems. Such an approach is denoted to as kernel trick. To this end, the inner product in a given algorithm is replaced by a kernel function K; then, utilizing the different K, one can find out which kernel function gives more expert outcomes.

2. LITERATURE REVIEW

The outlier detection task can be very time depleting and recently there has been an increasing interest in parallel/distributed methods for outlier detection.

Hung and Cheung et.al [1] presented a parallel version, called PENL, by the basic NL algorithm E. Knorr and R. Ng et.al. [2]. PENL is based on a definition of outlier employed in [2]: a distance-based outlier is a degree for which less than k points lie inside the distance in the input data set. This definition does not provide a ranking of outliers and needs to find an appropriate value of the parameter. Moreover, PENL is not suitable for distributed mining, because it requires that the whole data set is transferred among all the network nodes. Lozano and Acun et.al [3] proposed a parallel variant of the Bay’s algorithm S.D. Bay et.al. [4], which is based on a definition of distance-based outlier coherent with the one used here. However, the method did not scale well in two out of the four experiments presented. Moreover, this parallel version does not deal with the drawbacks of the centralized version in [4], which is sensitive to the order and to the distribution of the data set. Oteyetal.

In et.al [5] and Koufakou and Georgiopoulos et.al. [6] Proposed their strategies for distributed high-dimensional data sets. These methods are based on definitions of outlier which are completely different from the definition employed here, in that they are based on the concept of support, rather than on the use of distances. Dutta et al. [7] Proposed algorithms for the distributed computation of principal components and top-k outlier detection. In their approach, outliers are objects that deviate from the correlation structure of the data: A top-k outlier is an object having at most the kth largest sum of squared values in a specified number of the lowest order principal components, where each element is normalized to its departure. This definition neither implies nor is implied by the definition employed in this work. For instance, if all clusters are situated far from the mean of the data set, distance-based outliers close to the mean are not necessarily exceptional in the correlation structure. On the other hand, objects having large values in the first principal components need not have smaller weight than objects which vary from the correlation structure in the low-order elements.

Ramaswamy et al. [8] Modified the definition of outlier introduced by Knorr and Ng and consider as outliers the top n points p whose distance to their kth nearest neighbor is greatest. To detect outliers, a partition-based algorithm is presented that, first, partitions the input levels using a clustering algorithm and, then, prunes those partitions that cannot contain outliers. The experiments, up to 10 dimensions, show that the method scales well with regard to both data set size and dimensionality. This definition, however, does not take into account the information contained in the k-neighborhood of a point and, thus, it could not properly distinguish between a dense or sparse neighborhood

An analogous definition of outlier based on the k-nearest neighbors has been used in E. Eskin et.al. [10] for unsupervised anomaly detection to detect intrusions in unlabeled data. Data elements are mapped in a feature space and anomalies are detected by determining which points lie in sparse regions of the feature space. Experiments on data sets of network connections and system call traces showed that the algorithms were able to find intrusions over unlabeled data. More recently, Bay and Schwabacher et.al.[9], in order to find the top-n distance-based outliers of an input data set, augmented the naive distance-based nested loop algorithm, which finds the k nearest neighbors of each data set point, with a simple pruning rule and randomization obtaining a near linear scaling on real, large, and high-dimensional data sets. The algorithm is sensitive to the order and to the distribution of the data set. If the data is sorted or correlated, the operation could be miserable.

The Distributed Solving Set algorithm is different, since it computes the true global model through iterations where only selected global data and all the local data are required.

The core computation executed at each node consists in the following steps:

- Receiving the current solving set objects together with the current lower bound for the weight of the top nth outlier,
- Comparing them with the local objects,
- Taking out a new set of local candidate objects (the objects with the top weights, according to the current estimate) together with the list of local nearest neighbors with respect to the solving set and, in the end,
- Determining the number of local active objects, that is, the objects have weight not smaller than the current lower bound.

The comparison is performed in several distinct cycles, in order to avoid spare computations. The above data are used in the synchronization step by the supervisor node to generate a new set of global candidates to be used in the following iteration, and for each of them the true list of distances from the nearest neighbors, to compute the new (increased) lower bound for the weight.

The problem of solving set algorithm during the calculation, if an object becomes non-active, and then it will not be considered for insertion into the set of candidates, because it cannot be an outlier. As the algorithm moves to new objects, more precise weights are calculated and the turn of non-active objects expanses. The algorithm ends when no more objectives have found, i.e., when all the objects not yet got as candidates are non-active, and thus the candidate object becomes null. The solving set is the union of the sets candidate object computed during each iteration. Existing Line-up technique has just provided a rank to number of products. It didn’t optimize the ranking. Now, in this paper, overcome these entire problems and provide the best solution.

3. PROBLEM FINDING

In the present ranking system the specific attribute base ranking are As support considering several companies' data which are offering a user their own attractive terms, but still there are some hidden (outliers) which they never highlight to the user, but still if the particular user if interested on finding ranking of products availability on the basis of outlier attribute, person should able to find out the desired solution via a medium from a large number of dataset available.

As there might be some hidden charges or other related tax charges or annual maintenance or interest higher than the other might be present as outlier which might be helpful for the user to take action on selecting a rank. The same kind of problem can consider while selecting the multiple products for requirement or selecting medical or other equipment's. So today still here is a problem for the availability of such tool to gather or to observe the required ranking parameters.

3.1 Proposed Solution

Outlier detection always succeeds to get the other attributes which are not taken as consideration of the selected data set, find out the outliers multi attributes which are not considered as to provide the ranking to the user.

Here going to aim a tool to detect possible outlier attributes from the all attributes which are the part of product attribute while selecting the product.

On detecting attribute outliers going to get use of them to perform ranking on taking considered attribute as outlier and user choice attribute as a main attribute so that user can have a product ranking or a specific survey on his choice or demand.

The outlier technique is being used in product ranking optimization for the user choice. The visual ranking and outlier selection approach in this tool which gives us a better experience which using the special tool, because it has been seen that visual presentation is a far better choice than the statically approach which presenting any kind of data and attribute survey result in the market.

3.2 Proposed Algorithm

In order to perform task well & efficient going to perform various algorithms set for the results are solving set.

The Solving set algorithm for finding the outlier attributes which finding the desired attributes for the ranking optimization thoughtfulness. So in order to find disjoined non considered attributes going to apply the algorithm for finding the outliers and then we will store the outliers in a separate dataset for considering next algorithm usage.

The Lineup algorithm helps us to optimize the ranking on providing the attribute of choice for the consideration, but again line up gives us ranking of the particular provided attributes only, so here to perform HYBRID DSS. First, find all the relevant candidate data set in this data set, and then we will do HYBRID DSS. Visualization of the ranking will be executed and random choice option for selecting another attribute can be provided to the user for further ranking optimization.

Basic steps of hybrid DSS are as follows:

Step 1: Load data sets from the saved Database.

Step 2: Load Candidate data sets.

Step 3: Compare Candidate data set with the exiting original data set.

Step 4: Employ Hybrid algo to find Maximum no. of Outlier this will take minimum computation time as compared to Lazy Dss and DSS.

Step 5: Employ Outlier dataset over line-up algo.

Step 6: Now with the aid of multiple attribute over step 5 we best rank over the data set.

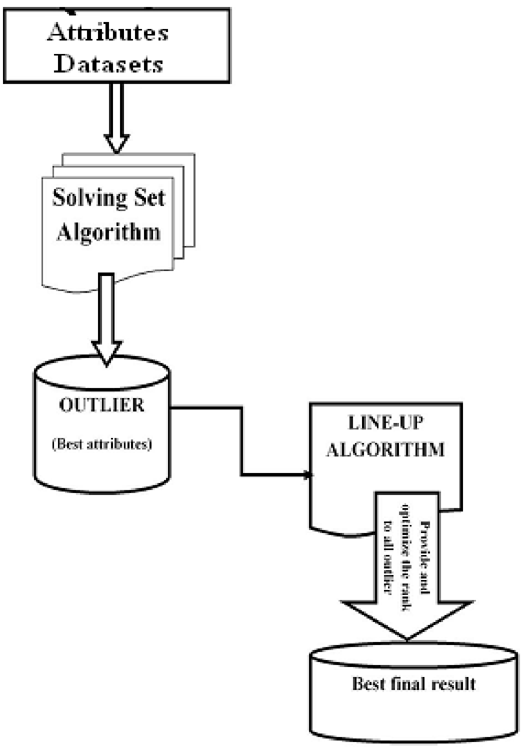

Figure 2. Ranking Optimization using Outlier Detection Technique

3.3 Solution Description Technique

As have discussed more about the existing problem domain here with the solution of the persisting problem with a solution for performing two algorithms which in this mentioned in the previous point for consideration. In this paper, doing the unique work on applying the hybrid DSS algorithm for the outlier detection and on getting the outliers can be perform optimization for ranking for the attributes.

APPLICATION FOR CONSIDERATION

In order to make sense of our work we are going to take multiple dataset from the various fields such as-

- Multiple credit card companies data and their attributes.
- Healthcare Insurance card dataset.
- Multiple plans available in different product or scheme (EMI Option or product category and purchasing plan) dataset.

Considering this above application in demand today going to perform experiment in these emerging are which further can be used by the user as an application for finding out their required solution on these application domains and problems.

4. CONCLUSION

The research work on the outlier has been used more for various applications mentioned on the founding part, here to function an algorithm from the outlier detection technique and further going to use the hybrid DSS algorithm for getting the best result in order to make use of the algorithm for ranking optimization and provide the best solution in competitive areas. So here conclude that this is one of the best research in the application of outlier and also in the area of to provide us better optimize and dynamic survey result according to the required on demand choice by proposed integrated tool.

5. REFERENCES

[1] E. Hung and D.W. Cheung, “Parallel Mining of Outliers in Large Database,” Distributed and Parallel Databases, vol. 12, 2002.

[2] E. Knorr and R. Ng, “Algorithms for Mining Distance-Based Outliers in Large Datasets,” Proc. 24rd Int’l Conf. Very Large Data Bases (VLDB), pp. 392-403, 1997.

[3] E. Lozano and E. Acun ˜a, “Parallel Algorithms for Distance-Based and Density-Based Outliers,” Proc. Fifth IEEE Int’l Conf. Data Mining (ICDM), pp. 729-732, 2005.

[4] S.D. Bay and M. Schwabacher, “Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule,” Proc. Ninth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), 2003.

[5] M.E. Otey, A. Ghoting, and S. Parthasarathy, “Fast Distributed Outlier Detection in Mixed-Attribute Data Sets,” Data Mining Knowledge Discovery, vol. 12, nos. 2/3, pp. 203-227, 2004.

[6] A. Koufakou and M. Georgiopoulos, “A Fast Outlier Detection Strategy for Distributed High-Dimensional Data Sets with Mixed Attributes,” Data Mining Knowledge Discovery, vol. 20, pp. 259-279, 2009.

[7] H. Dutta, C. Giannella, K.D. Borne, and H. Kargupta, “Distributed Top-K Outlier Detection from Astronomy Catalogs Using the DEMAC System,” Proc. SIAM Int’l Conf. Data Mining (SDM), 2007.

[8] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient Algorithms for Mining Outliers from Large Data Sets,” Proc. Int’l Conf. Managment of Data (SIGMOD ’00), pp. 427-438, 2000.

[9] S.D. Bay and M. Schwabacher, “Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule,” Proc. Int’l Conf. Knowledge Discovery and Data Mining (KDD ’03), 2003

[10] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, “A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data,” Applications of Data Mining in Computer Security, Kluwer, 2002.