Enhanced Outlier Detection for High Dimensional Data Using Different Neighbor Metrics

Shruti S. Rakhe1, Archana S. Vaidya2
GES’s RH Sapat COE, Krishi Nagar, Nashik

Abstract:
The task of identifying patterns that do not take after the built up customary conduct is termed as outlier detection. It has varied applications in number of fields like fraud detection, intrusion detection, abnormal weather detection, detecting unexpected entries in text data, etc. Majority of the aforementioned applications are concerned with high dimensional data. As the dimensions of data increases, performance of many traditional outlier detection methods deteriorates because of curse of dimensionality. Here a novel method for outlier detection is proposed. The proposed system uses reverse nearest neighbor along with shared nearest neighbor. Reverse nearest neighbor gives the count of number of times a particular object occurs among k nearest neighbors of all other points in a data set. Shared nearest neighbor finds similarity between points. Shared nearest neighbor when used as a secondary measure can beat the curse of dimensionality.

Keywords: curse of dimensionality, outlier detection, reverse nearest neighbor, shared nearest neighbor

I. INTRODUCTION

The famous definition of outlier as given by Hawkins is “An outlier/anomaly is an observation which differs so much from the rest of the observations so as to provoke suspicions that it was generated by a different method” [1]. Outlier discovery is the errand of deciding uncommon yet critical examples in information.

Outliers are vital as they might have important and significant information in several domains, such as intrusion and fraud detection, and medical diagnosis. An outlier, many a times comprises of valuable information about the unusual characteristics of a system, which may reflect the process of data generation. Moreover, outlier detection has various applications in numerous fields. Some of them are: to detect fraudulent applications for credit cards, to detect deceptive usage of credit cards or mobile phones, to detect deceptive applications or potentially troublesome customers, detecting unauthorized access in computer networks, to detect mobile phone deception by monitoring use of phone or doubtful trades in the equity markets, monitoring the performance of computer networks, to detect mislabeled data in a training data set, supervising medical condition - such as heart-rate monitors, detecting unexpected entries in databases which eventually detect errors for data mining, frauds or valid, yet unexpected entries.

Unsupervised, semi-supervised, and supervised, are the categories in which outlier detection methods can be broadly classified. Categorization depends on the availability of labels for outliers and/or regular instances. From these categories, unsupervised approaches are more widely applied [2], because the remaining categories require precise and illustrative labels that are not easily available. Unsupervised approaches include distance-based methods [3], [4], [5] that generally depend on a measure of distance or similarity and density based methods which consider density of an object’s neighborhood to detect outliers.

It is commonly seen that distances becomes futile [6], as a result of the “curse of dimensionality,” because distance measures concentrate, i.e., pairwise distances become unnoticeable as dimensionality increases [7], [8]. The consequence of distance concentration on unsupervised outlier detection was inferred to be that each point in high-dimensional space becomes an almost equally good outlier [9]. The objective of proposed system is to find outliers without considering contextual or collective information. To do this, unsupervised method is used. Ranking is done using reverse nearest neighbor [10] and shared neighbor distance [11].

In the following sections of paper, existing unsupervised methods for outlier detection are discussed in section II, the design of proposed system in described in section III, implementation details are provided in section IV and finally results are presented in section V. Paper concludes in section VI.

II. DESIGN OF PROPOSED SYSTEM

Proposed system can be used in both high and low dimensional data. Figure 1 shows the architecture of outlier detection system.

Figure 1: Proposed System

As it can be seen from figure that the dataset is passed to RNN (Reverse Nearest Neighbor) algorithm and SNN(Shared Nearest Neighbor) algorithm and then we get the outliers. Now we will go through the algorithms one by one.

a) Reverse Nearest Neighbor:

It is generally believed that in high-dimensional space unsupervised methods identify every point as an outlier, since distances become unnoticeable as dimensionality rises [9]. In [21] this belief was contested by proving that the exact opposite may take place: as dimensionality increases, outliers produced by various techniques from the data tend to be detected as more pronounced by unsupervised methods, assuming all dimensions have useful information. Distance concentration is the tendency of distances in high-dimensional data to become almost unnoticeable as dimensionality increases, and is generally portrayed through a
ratio of a notion of spread (e.g., standard deviation) and magnitude (e.g., the expected value) of the distribution of distances of all points in a data set to some reference point. It is said that distances concentrate, if this ratio tends to 0 as dimensionality goes to infinity. When random data with its coordinates and Euclidean distance is considered, concentration is revealed in the fact that, the standard deviation of the distribution of distances remains constant, as dimensionality increases, though the mean value continues to grow. More visually it can be said that, as dimensionality increases, all points tend to lie approximately on a hyper sphere centered at the reference point, whose radius is the mean distance [10]. Any point can be used as the reference point in high-dimensional space, producing the concentration effect: the radius of the sphere (the expected distance to the reference point) increases with dimensionality, while the spread of points above and below the surface (e.g., the standard deviation of the distance distribution) becomes negligible compared to the radius. Occurrence of anihubs is facet of the “curse of dimensionality” related to distance concentration. This facet will be generally referred to as hubness [22]. To illustrate hubness, the concept of k-occurrences, hubs (nearest neighbors) and antihubs (reverse neighbors) is defined.

1) k-occurrences: Let D ⊆ R^d be a finite set of ‘n’ points. For point x ∈ D and a given distance or similarity measure, the number of k-occurrences, denoted N_k(x), is the number of times x occurs among the k nearest neighbors of all other points in D. Equivalently, N_k(x) is the reverse k-nearest neighbor count of x within D.

2) hubs and antihubs: For q ∈ (0, 1), hubs are the [nq] points x ∈ D with the highest values of N_q(x). For p ∈ (0, 1), p < 1- q, antihubs are the [np] points x ∈ D with the lowest values of N_p(x).

3) probability: Let functions p_{1,k} where i, k ∈ {1, 2, ..., n}, be defined as:

\[ p_{1,k}(x) = \begin{cases} 1, & \text{if } x \text{ is among the } k \text{ nearest neighbors of } x_i \text{ according to } dist \\ 0, & \text{otherwise} \end{cases} \tag{1} \]

In equation (1) p_{1,k} denotes the probability that a particular object in dataset is present in the kNN list of other object in the dataset. And depending on the probability calculation value of N_k(x) is calculated.

The presence of antihubs is closely connected with outliers both in high-dimensional and low dimensional data. So the Reverse Nearest Neighbor is based on this property.

Natural outlier scoring based on N_k counts was used in the ODIN (Outlier Detection using Indegree Number.) method [20]. Proposed method defines the outlier score of object x from data set D as a function of N_k (x), and is given in Algorithm 1(RNN):

Temporary variables: t

Steps:
1) For each i ∈ {1, 2, ..., n}
2) \[ N_k(x_i) = \sum_{j=1}^{d} p_{1,k}(x) \]
3) \[ t = N_k(x_i) \]

The variable t in the algorithm gives the count of how many times a particular object is present in the neighborhood list of all other objects in the dataset.

b) Shared Nearest Neighbor

Shared neighbor distances are used as secondary distance measures to deal with high-dimensional data. Similarity between points is given as the number of shared neighbors in their k-neighbor sets. Shared neighbor distances have been considered as a potential remedy for the curse of dimensionality [24]. This secondary measure is based on the rankings induced by a specified primary similarity measure (Algorithm 1).

A common approach to shared neighbor distances is to count the number of shared nearest neighbors (SNN) among the pairs of points for a specified, fixed neighborhood size. Let D = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) be the data set, where x are the objects in particular dataset and y_i, i=1,2,...,n are the labels. The k-neighborhood of x_i is denoted by D_k (x_i). A shared neighbor similarity between two points is then usually defined as:

\[ \text{simcos}_s(x_i, x_j) = \frac{|D_k(x_i) \cap D_k(x_j)|}{s} \tag{1} \]

where s denotes the neighborhood size. Points which are dissimilar from majority of the objects in the dataset are considered as outliers.

III. MATHEMATICAL MODELLING

- S= {I, O, F, U, A}
- I: Input: D (Where,
- D=Ordered data set (x_1, x_2, ..., x_n)
- ao1, churn, ctg3, ctg10, kdd999, kdd99, mammography, nba-allstar-1951-1972, nba-allstar-1973-2009, thyroid-sick, us-crime, wilt, are dataset. Any one of the datasets can be given as input to system
- O: Output: {s} (Where,
- s= {s_1, s_2, ..., s_n}, where s_i is the outlier
- F: Functions: (Reverse Nearest Neighbor, Shared Neighbor Distances)
- A: Assumption: {a,b,c}
- a: all (or most) data attributes are meaningful, i.e. not noisy
- b: It is assumed that all attributes carry useful information
- c: It is assumed that only alpha numeric values are given as input

Reverse Nearest Neighbor, denotes the outlier score of point x from data set D as a function of N_k (x) Shared Neighbor Distances is a similarity measure which is used to improve the ranking produced by Reverse Nearest Neighbor.

- U: defines user who wants to find outlier.

In the following section we will see the implementation of Reverse Nearest Neighbor and Shared Neighbor Distances.

IV. IMPLEMENTATION

Proposed system is implemented in .net framework. Datasets used are mentioned in mathematical model in section III, now we will see the details.

1. aloi dataset contains 65 dimensional 50,000 points [25], [26]
2. churn has 5,000 point which 18 dimensional [27], [28]
3. ctg3 and ctg10 contain 36 dimensional 2,126 points each [29]
4. wilt is 6 dimensional dataset having 4,839 points [30]
5. kdd999-r2l and kdd99-u2r has 39 dimensional 68,338 and 67,395 points respectively [30]
6. mammography has 11,183 points having 7 dimensions [31]
8. nba-allstar-1973-2009 has 18 dimensional 16,916 points [32]
9. thyroid-sick consists of 3,772 points with 53 dimensions [31]
10. us-crime has 1,994 points with 101 dimensions [31]

All of these are high dimensional datasets.

V. RESULTS
In figure 1 csv file represents the datasets discussed above, N represents number of dimensions of dataset, D represents dimensions or number of columns in dataset and ES_Outlier gives the percentage of outliers detected which is shown graphically in figure 2.

As it can be seen in the figure 3, the output of proposed system is plotted on the graph. X axis in the graph represents the dataset which are discussed above and Y axis represents the percentage of outliers detected from dataset by proposed system. From figure 2 we can clearly see that proposed system (RNN+SNN) detects more outliers (yellow bar graph) as compared to existing system (blue bar graph in figure 2).

Moreover figure 4 shows the time taken by proposed system (yellow bar graph) and existing system (blue bar graph) X axis shows the id of datasets and Y axis shows the time taken by those datasets in milliseconds. In proposed system we are traversing dataset from first object as well as last object simultaneously. So as we can clearly see in the graph time taken by proposed system is less as compared to existing system which is traversing dataset from starting object only (in one direction).

Performance of the proposed system can be measured using precision and recall if the count of correct number of outliers present in the dataset is known.

VII. REFERENCES


[15] Lei Cao, Di Yangt, Qingyang Wang, Yanwei Yu+, Jiayuan Wang, Elke A. Rundensteiner, “Scalable Distance-Based Outlier Detection over High-Volume Data Streams” in ICDE Conference 2014 IEEE.


