DATA CLEANING-DISCOVERING NULL AND OUTLIERS

Bavneet Singh
bs3485@nyu.edu

Abhichandra Ravi
ar5439@nyu.edu

Divyanshu Singh
ds5387@nyu.edu

ABSTRACT

The quantity of data produced every day and reliance on these datasets by organizations is at a exponential increase. Unfortunately, this huge amount of derived data is not meaningful and does not provide any insights. The data usually contains duplicates, bad-formatting, incorrect and incomplete data etc. Therefore, making sense of this immense data is one of the biggest challenges we face today. The goal of this project is to mine the “raw data” for outliers and NULL values, and possibly conduct an analysis of the outliers.

KEYWORDS

Outliers, NULL, K-means, BK-means

1. INTRODUCTION

The quantity of data produced every day and reliance on these datasets by organizations is at an exponential increase. Data warehousing is one the major technologies that seek to create a repository from all the sources from which they obtain their data. This helps the organization get their information as it can be accessed and used from one point. Companies over the last couple of decades have done more data logging and capturing with the advent of computers with database capabilities. Many have found that these data are quite useful to augment or focus market groups if only the information was available for statistical analyses. Unfortunately, this huge amount of derived data is not meaningful and does not provide any insights. The data usually contains duplicates, bad-formatting, incorrect and incomplete data etc. Therefore, making sense of this immense data is one of the biggest challenges we face today.

2. PROBLEM FORMULATION

Data cleaning, which is also known as data cleansing is the process of detecting and removing errors and inconsistencies from data in order to improve the quality of the dataset. The above-mentioned data quality problems are usually present in files and databases. Usually, multiple data sources are integrated in data warehouses, federated databases systems etc, and the need for data cleaning is highly significant. Therefore, in order to provide access to accurate and consistent data, consolidation of these different data representations and the process of eliminating duplicate and irrelevant information become necessary. The classification of data quality problems can be divided into two main categories: single-source and multiple-source problems. At the single-source, they are into schema level and instance level related problems without considering the occurrence in a single relation. The single-source problems deal with attribute, record, record type and source whereas the multiple-source problems deal with naming conflicts, schema-level conflicts and the identification of overlapping data which refers to same real-world entity.

3. METHOD AND DESIGN

The objective of the proposed algorithm that we call outlier removal clustering (ORC), is to produce a codebook as close as possible to the mean vector parameters that generated the original data. It consists of two consecutive stages, which are repeated several times. In the

first stage, we prepare the data by removing Null values, performing case normalization, trimming space on both ends of string, in case we want to keep null values we can fill it with dummy variable. We then perform K-means algorithm until convergence, and in the third stage, we assign an outlyingness factor for each vector. Factor depends on its distance from the cluster centroid. Then algorithm iterations start, with first finding the vector with maximum distance to the partition centroid *dmax*.
Before performing K-means we need to perform additional changes to the dataframes:

A. **Removal of NULL Values**
B. **Identifying NULL Values**
   Identifying and filling NULL values with one default value instead of removing it.
C. **Removal of Duplicate Values**
D. **Removal of Duplicate Values**
   Removal of leading and trailing whitespaces.
E. **Case Normalization**
   Converting the content in columns in upper or lower case.
F. **StringIndexer**
   StringIndexer encodes a string column of labels to a column of label indices. The indices are in [0, numLabels), ordered by label frequencies, so the most frequent label gets index 0.
G. **OneHotEncoding**
   It maps a column of label indices to a column of binary vectors, with at most a single one-value.
H. **VectorAssembler**
   VectorAssembler is a transformer that combines a given list of columns into a single vector column. It is useful for combining raw features and features generated by different feature transformers into a single feature vector, in order to train ML models.
I. **Principal Component Analysis(PCA)**
   A PCA class trains a model to project vectors to a low-dimensional space using PCA.

Then the result from the PCA is sent to K-means and BK-means outlier detection algorithm.

3.1. ALGORITHM

The algorithm which has been used in-order to identify outliers is the Distance Based Algorithm. In this approach, we distribute the process of finding the outliers into three phases. Initially we run k-means clustering algorithm to find k cluster, then calculate accuracy and WSSE index of K-means clustering. The three phases involved are as follows:

1st Phase: In the first phase, we find threshold value. This is obtained by calculating the pairwise distance for whole dataset. Given an mx by n data matrix X, which is treated as mx (1 by n) row vectors x_1, x_2... x_{mx}, and my-by-n data matrix Y, which is treated as my (1-by-n) row vectors y1, y2...y_{my}, the various distances between the vector x_s and y_t are defined as follows:

Euclidean distance: $d_{2st} = (x_s - y_t)\,(x_s - y_t)$

Then, we take the maximum and minimum value of pairwise distance for all observation and calculate the threshold value.

Threshold value = (maximum distance + minimum distance) ÷ 2

2nd Phase: Once we have obtained the Euclidean distance of all data in the dataset, we compare these values with the threshold value obtained in the first phase.
If **Distance > Threshold** value this data is considered as outlier.
If **Distance ≤ Threshold** value this data is not outlier.

3rd phase: Then we compare the outliers we found with the result of another clustering model BK-means. The outlier detection for BK-means is similar to K-means. We label the tuples as outliers only if both the clustering algorithm agree on it.

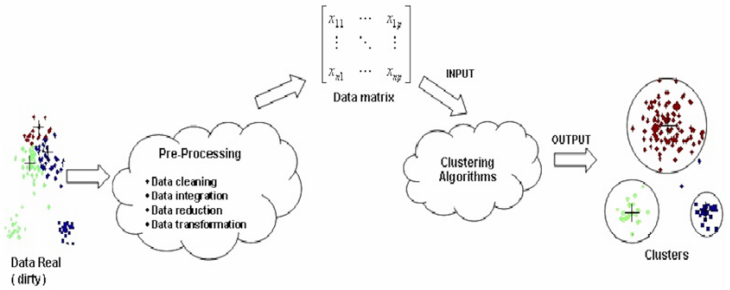

Fig. 3.1.1

3.2. ARCHITECTURE

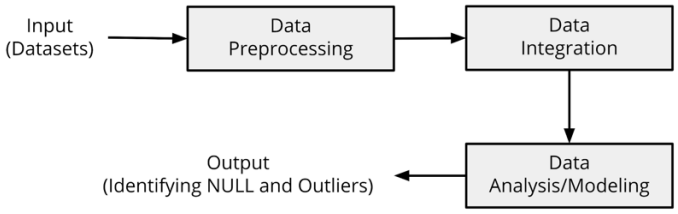

Fig. 3.2.1

4. TECHNICAL DEPTH AND INNOVATION

What is an Outlier? An outlier is defined as a noisy observation, which does not fit to the assumed model that generated the data. In clustering, outliers are considered as observations that should be removed in order to make clustering more reliable. In outlier detection methods based on clustering, outlier is defined to be an observation that does not fit to the overall clustering pattern. The ability to detect outliers can be improved using a combined perspective of outlier detection and clustering. Some clustering algorithms, for example K-Means and Bisecting K-Means, handle outliers as special observations, but their main concern is clustering the dataset, not detecting outliers. K-Means is a local density-based outlier detection algorithm. Local density-based scheme can be used in cluster thinning. Outlier removal algorithm can remove vectors from the overlapping regions between clusters, if the assumption holds that the regions are of relatively low density. Higher density is found near the cluster centroid. An obvious approach to use outlier rejection in the cluster thinning is as follows: (i) identify outliers (ii) Cluster the data using any method.

Here we cluster the data using K-means and bisecting K-means, we first establish a threshold for determining the outlier. This threshold, in our case is minimum distance from the cluster centroid + maximum distance from the cluster centroid divided by two. If the distance of the particular tuple is greater than the threshold i.e it lies beyond the threshold will be treated as Outliers. We make a set *S1* of all the tuples identified as an outlier. This process is repeated with bisecting K-mean and generate a set *S2.* Then the outliers are finally determined by taking an intersection of the set *S1* and *S2*.

After determining the outliers, we just add an extra column in the data set indicating whether this particular is an outlier or not. The purpose of not dropping the outliers is that they can be further used to find out certain correlations or patterns. For example, if we analyze the credit card usage of the particular user and we find outliers in that data, there is a high probability that it might be the case of a stolen credit card.

5. RESULT

The final output of the algorithm is parsed into an excel file which can be checked in the GitHub link provided in the folder output.

6. CODE REPOSITORY:

https://github.com/bavneetsingh16/Big-Data-Project

REFERENCES

[1] Ivezić, Z., Connolly, A., Vanderplas, J. T., & Gray, A. (2014). *Statistics, data mining, and machine learning in astronomy: A practical Python guide for the analysis of survey data*. Princeton, NJ: Princeton University Press.

[2] Chandola, V., Mithal, V., & Kumar, V. (2008). Comparative Evaluation of Anomaly Detection Techniques for Sequence Data. *2008 Eighth IEEE International Conference on Data Mining*. doi:10.1109/icdm.2008.151

[3] Yu, D., Sheikholeslami, G., & Zhang, A. (2002). Find Out : Finding Outliers in Very Large Datasets. *Knowledge and Information Systems, 4*(4), 387-412. doi:10.1007/s101150200013

[4] Hautamäki, Ville, Svetlana Cherednichenko, Ismo Kärkkäinen, Tomi Kinnunen, and Pasi Fränti. "Improving K-Means by Outlier Removal." *Image Analysis Lecture Notes in Computer Science*, 2005, 978-87. doi:10.1007/11499145_99.

[5] Anwesha Barai (Deb) , Lopamudra Dey (2017). Outlier Detection and Removal Algorithm in K-Means and Hierarchical Clustering. World Journal of Computer Application and Technology, 5 , 24 - 29. doi: 10.13189/wjcat.2017.050202.

[6] "Machine Learning Library (MLlib) Guide." Apache Spark™ - Unified Analytics Engine for Big Data, spark.apache.org/docs/latest/ml-guide.html.