DSGA-1004: Project Proposal

Isaac Haberman, Joe Sloan
3/18/2018 DSGA-1004

Choice of Project: Given a data set collection, implement functions to discover outliers and NULL values in the different data sets

Outlier discovery and anomaly detection are among the most researched fields in information theory, statistics, data science and other similar disciplines[1]. As such, there are a plethora of techniques and algorithms to choose from when trying to design a solution. Chandola, Banerjee and Kumar's[2] survey of anomaly detection designate six popular classes of techniques: classification algorithms, K-nearest neighbors approach, clustering techniques, statistical methods, information theory techniques, and spectral methods. Of those six broad classes, not all are applicable to all problems; classification algorithms require the use of labeled data and statistical methods notably require the assumption of a knowable underlying distribution. Some techniques also have particular challenges based on the data being used, such as the decrease in effectiveness of distance-based clustering techniques as dimensionality of a dataset increases. This makes it difficult to choose one particular technique for outlier detection across a wide variety of datasets.

Chandola, Banerjee and Kumar's survey also classifies outliers into three broad types: point, contextual, and collective. Point outliers are the singular point anomalies that one typically thinks of when considering outliers, making them the easiest to detect. Contextual outliers are points that do not make sense given outside information about the data (they are outliers after data conditioning), and collective outliers are groups or clusters of similar anomalous instances. Time-series data also presents unique challenges, though they can be framed as a subset of the larger classification of contextual outliers.

Given the above results, it is clear that no one outlier detection technique will serve every possible need. For our project, we propose a suite of functions built for Spark in Python, which utilize several of the above methods to classify broad ranges of outliers. As specified above, we will treat NULL values as a subset of outliers to be flagged. A user will input the dataset, a specified sensitivity in the form of a percentage estimate of outliers[3], and any settings specific to the functions specified with the ability to include or exclude certain sets of techniques. Given those inputs, the functions will output the data with the outliers removed or marked as specified by the user and based on the output of all the functions run.

Our suite of functions will utilize the techniques we find most applicable to big data while offering ways to maximize the types of outliers it can detect. From our research, we believe that the greedy entropy-based method proposed by He, Xu, and Deng[4] will be appropriate for the classification of both point and contextual outliers in big data contexts. To our knowledge, their algorithm has not been tested on large data-sets, however it has a relatively quick runtime, iterating through the data once per assumed outlier instance. As written, their technique applies to each feature independently. In our implementation, we can opt to include or exclude entropy from other features when considering

[1] We assume NULL values are either entered as outlier's values like -99999 or are missing data, which can be found without the use of an advanced algorithm

[2] http://www.dtc.umn.edu/publications/reports/2008_16.pdf

[3] A technique used in many of our listed papers

[4] https://arxiv.org/ftp/cs/papers/0507/0507065.pdf

each example based on the user's preference. As an alternative for point and contextual outliers, we will also investigate the distance-based algorithms detailed by Ramaswamy, Rastogi, and Shim[5], which have demonstrated efficiency on larger datasets. Lastly, we will combine clustering techniques and the works of Olsson and Holst[6] to handle collective outliers.

We will test the effectiveness of our solution on known datasets used for outlier classification that we will aggregate from several sources. Stony Brook University's Department of Computer Science maintains a collection of outlier detection datasets and classification rates with various techniques[7] that we can use for this purpose. Specifically, the Mulcross, ForestCover, and HTTP datasets at Stony Brook are large enough to be a meaningful test for our solution in a big data context. Additionally, Netflix has published a climate dataset[8] that they use to test their outlier detection algorithms, which are certainly on the scale of big data. Finally, the Numenta Anomaly Benchmark is a Github repository of known datasets and performance benchmarks for testing anomaly detection algorithms. We will compare our solution's performance to published results on each of these benchmark datasets in speed, accuracy, and compute efficiency. If time permits, we can also attempt to replicate those published results for comparison purposes.

[5] http://ftp10.us.freebsd.org/users/azhang/disc/disc01/cd1/out/papers/sigmod/efficientalgorisrrak.pdf
[6] https://pdfs.semanticscholar.org/e74e/37e6fc1c5ad030ad1a553193034ff3afbd8f.pdf
[7] http://odds.cs.stonybrook.edu/
[8] https://medium.com/netflix-techblog/rad-outlier-detection-on-big-data-d6b0494371cc