Marcus Lehr
Veer Kumar

Project Checkpoint #3

Proposal / Implementation Plan

Project Introduction

Overview

For our project, we have chosen to investigate a novel univariate outlier detection algorithm previously created by one of us. It was created in the context of quantitative data being captured in clinical studies to ascertain data points that warrant manual review in real-time while studies are ongoing. However, while the algorithm was determined to work well empirically, its properties have not been formally investigated.

Detecting outliers holds substantial significance in the field of medicine, as these data points carry significant and often crucial information (Chandola et al., 2009). In healthcare, techniques for identifying outliers are employed to spot unusual patterns in patient records, which can potentially contain valuable information such as emerging disease symptoms. To aid in these efforts, we hope to contribute a new tool to the arsenal of outlier techniques by more thoroughly investigating the utility of our method and comparing it with other established methods.

Problem description

When conducting human studies the data collected are inherently precious. This is because human studies are more expensive compared to studies with model organisms and there are regulatory pressures to keep sample sizes as small as possible for humane reasons. Because of these reasons, having to remove subjects from the analysis due to poor data quality is a major concern. In order to address this, we have implemented data quality checks with the goal of flagging outliers for manual review. These checks are done in real time during the study to improve the odds that invalid data points can be recovered or re-measured. Data recovery becomes more difficult or impossible as time passes due to the time sensitivity of certain types of data and the risk of losing contact with a participant after the study is over. Additionally, while the quality control checks should be biased against false negatives, too many false positives will burden study staff and participants. Thus, a requirement of our algorithm is that it is resilient to small sample sizes and different distribution characteristics. Robust performance at small sample sizes allows monitoring to begin as soon as possible after the study starts. Furthermore, the algorithm must be highly generalizable. Each study collects different types of data and samples different populations of interest and thus the underlying distribution characteristics for any given data measure cannot be predicted in advance.

To meet these requirements, a 1-dimensional algorithm is most suitable. Many simple outlier methods exist for this type of data, typically relying on the data distribution characteristics. Some common methods include using a Z-score or a multiple of the IQR range as a thresholding mechanism. Initially, such methods were explored but found to be inadequate for multiple reasons. First was that choosing an appropriate threshold to maximize accuracy proved to be challenging. Second, the ideal threshold is different for different types of data and determining that threshold requires prior knowledge about distribution of the data. Finally, distribution characteristics themselves are sensitive to outliers and consequently very large outliers can mask more subtle ones. Conversely, outliers based simply on distribution characteristics are prone to false positives. For example, if using a Z-score threshold, the Z-scores will change along with the standard deviation of the data. The standard deviation will contract as extreme values are removed and raise the Z-scores of the remaining data. The trouble is that when empirically defining a distribution, the data used to create that definition will often contain data at the fringes of it.

Algorithm description

To address this problem, we developed an algorithm which looks for changes in the distribution rather than using a distribution metric to create a threshold. The algorithm first generates normal quantiles for the data- which would typically be used to create a quantile-quantile (QQ) plot- and fits a linear regression then captures its slope. Next, data points are ranked by their absolute Z-score and the highest ranked (most extreme) data point is removed. This process is repeated until all the data points have been removed. The absolute difference in slope after each removal is then calculated and a linear model is created against the number of points that have been removed. Cook's distance is then calculated to determine the influence of each data point on the model. Any data point which has a Cook's distance higher than the mean Cook's distance is flagged as an outlier. Finally, because data points become highly influential as the remaining sample size decreases, only the initial outlier flags are kept. Once a data point fails to meet the outlier criteria all remaining values, i.e. those with a lower Z-score, are ignored.

Proposed work

While this algorithm was found to produce satisfactory results in practice the statistical foundations remain ambiguous. Here we propose investigate the assumptions and limitations of the algorithm by generating synthetic data sets of different size and distribution types. Outliers will be injected by sampling different distributions. We will then use bootstrapping to determine the accuracy of outlier detection at various sample sizes. We will then investigate the performance characteristics of the algorithm. Finally, we will follow up on this by using the same methodologies to compare our algorithm to other known outlier algorithms that are applicable to univariate data. We will also apply it to real datasets to evaluate the usability of it to different data types.

Methods Justification and Data Exploration

Methods Justification

Firstly, we will seek to evaluate the algorithm's accuracy using synthetic datasets. Using synthetic data will allow us to easily assess the utility of the algorithm when exposed to data from different distributions so we can better understand the generalizability of the algorithm. Accuracy will be assessed using methods commonly applied to binary classification models, which largely consists of confusion matrix derived metrics (Agrawal, 2023; Vujović, 2021). A confusion matrix describes the relationship between actual classes and predicted classes (Table 1).

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Table 1. Confusion matrix. TP = True positives, TN = True negatives, FP = False positives, FN = False negatives

To further interpret the confusion matrix counts, we will look at several common metrics derived from these numbers (Table 2). Of particular importance to us will be overall accuracy and sensitivity, because the algorithm should be as accurate as possible while prioritizing the minimization of false negatives. Some number of false positives will lead to additional burden, but missing values that could be erroneous are of higher consequence in this context. The confusion matrix and relevant derivative metrics can be calculated in R using the associated functions in the caret package.

Metric	**Formula**
Accuracy	(TP + TN) / (TP + FP + TN + FN)
Sensitivity (aka True Positive Rate (TPR) or Recall)	TP / (TP + FN)
Specificity	TN / (TN + FP)
Positive Predictive Value (PPV)	TP / (TP + FP)
Negative Predictive Value (NPV)	TN / (TN + FN)

Table 2. Metrics derived from confusion matrix.

Another common classifier metric is the ROC AUC curve. Specifically, in our case it would be useful to look at precision-recall curves, as these are known to be more informative than standard ROC AUC curves for binary classification problems with class imbalances which are inherent to anomaly detection (Vujović, 2021). However, these curves are used to evaluate models which output a class probability. Our novel outlier method has a binary output and thus such metrics cannot be made without making modifications to the algorithm. While modifying the algorithm is certainly possible, it would require adding a parameter. The algorithm is currently parameters-less and this is desirable for the sake of generalizability and consistency. As a substitute for ROC curves will use the Kappa statistic, which measures the improvement of a model relative to random classifier (Vujović, 2021).

After performing the above analysis on the entire data sets, we will investigate the robustness of the algorithm to small sample sizes by repeating the analysis on datasets of different size. This will be done via bootstrapping at various N values using the bootstrap package in R. The expected result will be a distribution for the confusion matrix metrics described above at each value of N. This can be used to generate maximum likelihood estimates (MLEs) and confidence intervals (CIs). The goal will be to use these results to assess the minimum number of samples needed for reasonable performance. A note about this method is that we often expect the number of outliers to be zero so the algorithm should be quite robust to that environment. So particularly when generating smaller datasets, we will not include outliers in every iteration of the bootstrapping process but rather at some rate. Furthermore, when outliers are included, they will only exist as a rather small proportion of the data to reflect the rarity seen in real data. Additionally, while bootstrapping we can collect run times to assess the performance of the algorithm. This can be done in R using the built-in system.time() function. These run times can be compared across sample sizes to determine if it's consistent with the theoretical complexity determined via code analysis (Mejia, 2020).

Finally, we will compare these performance metrics to other comparable outlier detection methods. To begin with we will pit it against simpler statistical methods such as Pierce's criterion which assumes normal distribution and has an advantage since it can find and remove more than one outliers, other methods include Chauvenet's criterion, Z-score, or Mean Absolute Deviation (MAD) (Seo, 2006). We would also like to explore a novel method called Density peak clustering since it can be applied to multidimensional datasets.

Marcus Lehr
Veer Kumar

Project Checkpoint #3

Data Description

We will generate some synthetic data sets from different distributions and inject outliers to determine the algorithms robustness to different data distribution types. To make these datasets we will first begin by sampling a distribution using the suite of ‘r’ functions such as rnorm() function and then generate a uniform distribution over a larger range using runif() that aren’t included in the normal distribution and these can be the outliers for our dataset. Using synthetic data provides a controlled environment for evaluating our outlier detection algorithm.

In addition to this, we also look to test the generalizability of our outlier detection algorithm using real-world datasets. For instance, we are particularly looking to explore The National Health and Nutrition Examination Survey (NHANES) datasets have a wide range of data types includes demographic, socioeconomic, dietary, and health-related datasets and is known to be noisy.

Project Timeline

TASK	COMPLETION DATE
Background research for topic and methodology	Done
This schedule due	Nov 3
Sync meeting. Define synthetic datasets and test accuracy measures	Nov 8
Problem Set 4 due	Nov 10
Create and execute bootstrapping methods	Nov 13
Writeup of initial results due	Nov 17
Comparing these performance metrics to other comparable outlier detection methods	Nov 24
Work on PS 5 (dates are up to you, but make sure you leave time for this!)	Nov 26- Dec 1
Problem Set 5 due	Dec 1
Test algorithm on a real-world dataset	Dec 5
Final project writeup due	Dec 8
In-class Poster Session	Dec 11

References

Abraham, S. M. (2013). A Review of Class Imbalance Problem. *Journal of Network and Innovative Computing*, 332-340.

Agrawal, S. K. (2023, 09 29). *Metrics to Evaluate your Classification Model to take the right decisions*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

Ali, M. U. (2017). Using PCA and Factor Analysis for Dimensionality Reduction of Bioinformatics Data. *International Journal of Advanced Computer Science and Applications(IJACSA)* .

Chandola, V. &. (2009). Anomaly Detection: A Survey. *ACM Comput. Surv.*, 10.1145/1541880.1541882.

Haizhou Du, S. Z. (2016). Novel clustering-based approach for Local Outlier Detection. *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 802-811). San Francisco, CA: IEEE.

Maas, Y. (2019). Outlier detection in non-Gaussian distributions. *Bachelor's Thesis*. TU Delft.

McInnes, L. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimensionality Reduction. *arXiv preprint arXiv:1802.03426*.

Mejia, A. (2020, 10 03). *How to find time complexity of an algorithm?* Retrieved from adrianmejia: https://adrianmejia.com/how-to-find-time-complexity-of-an-algorithm-code-big-o-notation/

Noto, K. (2014). CSAX: Characterizing Systematic Anomalies in eXpression Data. *RECOMB*.

S. Papadimitriou, H. K. (2003). LOCI: fast outlier detection using the local correlation integral. *Proceedings 19th International Conference on Data Engineering* (pp. 315-326). Bangalore, India: IEEE.

Seo, S. (2006, 04 26). A Review and Comparison of Methods for Detecting Outliers. *Masters Thesis*. University of Pittsburgh.

Vujović, Ž. Đ. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, Volume 12, Issue. 6.

Wang, Y. (2005). Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*.