Data Manipulation and Detection

Tuvshin Selenge
Immatrikulationsnummer: 11815791
e11815791@student.tuwien.ac.at

Main supervisors:
Assoz.Prof. PD Dr. Jakob Müllner, Wirtschaftsuniversität Wien
Co-supervisor:
(Not decided yet), Technische Universität Wien

Domain-specific lecture: 4366 Foundations of International Business (WU Wien)

1 Abstract

This project investigates methods to detect data manipulation in datasets, addressing both accidental and deliberate alterations that can lead to practices such as p-hacking. Using unsupervised anomaly detection and statistical approaches, including z scores, cell-wise detection, isolation forest, and local outlier factor, the aim is to identify manipulated values at a granular level. In a collaborative experiment conducted by two students, independently altered datasets will be exchanged to cross-validate these detection methods on unseen data. Furthermore, the project seeks to reverse engineer manipulation processes, thereby enhancing data scrutiny and reinforcing data integrity in empirical research.

2 Introduction

Data manipulation is an increasing issue, particularly in research contexts. Researchers may unintentionally or intentionally manipulate data to generate novel or seemingly groundbreaking results, leading to problematic practices such as p-hacking. Our aim is to identify reliable methods for detecting manipulated data. This experiment will be conducted collaboratively: my colleague (Poj Netsiri) and I will independently manipulate datasets and then apply detection methods on each other's manipulated data to evaluate the effectiveness of our detection techniques.

3 Project Overview

3.1 Planned Workflow

1. **Data Collection:**
   We will utilize the ”cars” dataset provided by David K. Smith, which contains both numerical and categorical variables [2].

2. **Data Analysis:**
   We will perform statistical analyses, generate descriptive summaries, and create visualizations to develop an initial understanding of the data.

3. **Data Manipulation:**
   We will formulate three hypotheses and manipulate our datasets accordingly—either by altering values or structures—to artificially support these hypotheses. The manipulated datasets will then be exchanged with a collaborating colleague performing a parallel experiment.

4. **Application of Detection Methods:**
   After the data manipulation, we will apply selected detection methods (detailed in subsequent sections) to identify the alterations within each exchanged dataset.

5. **Evaluation of Detection Methods:**
   Finally, we will evaluate the effectiveness of the detection methods by thoroughly analyzing and visualizing the results from both datasets.

To smoothly transition from understanding the data to applying detection methods, after the initial analysis we deliberately manipulated the datasets as per our formulated hypotheses.

4 Methods to Detect and Determine Data Manipulation

4.1 Detecting Data Manipulation

Since we are working with unseen data, we will rely on unsupervised models and statistical approaches. Detecting data manipulation is as challenging as identifying outliers because manipulated values deviate from the true ones. As described by Raymaekers and Rousseeuw in their paper [1], detecting outliers at the cell level is not trivial. Therefore, we will explore the following methods:

1. **Statistical Approaches:**
   Using z-scores and distributional analysis, this method aims to flag marginal outliers or manipulations in multivariate numerical data.

2. **Cellwise Detection:**
   This approach treats each cell in the data matrix as potentially contaminated. The model is represented as:

   $$X = (I - B)\,Y + B\,Z,$$

   where:

   - X is the observed value in a cell,

- I is the identity matrix,
- Y is the true (uncontaminated) value,
- Z represents the contaminating (noise) value, and
- B is an indicator drawn from a Bernoulli distribution (with $B = 1$ indicating a contaminated cell and $B = 0$ otherwise).

This framework allows individual cells to be contaminated without necessarily affecting entire rows. An R package, `cellWise`, has been developed to implement this model [1].

3. **Isolation Forest:**
   Isolation Forest is an ensemble method for anomaly detection [3]. The algorithm builds trees by randomly selecting a feature and a split value to partition the data into subgroups. Anomalies are typically isolated quickly because they do not belong to dense clusters. In other words, if an anomalous point is separated from the majority within just a few splits—resulting in a short path length—it indicates abnormality. Normal points, which are part of larger, denser clusters, require more splits to isolate, resulting in longer path lengths.

4. **Outlier Detection with LOF (Local Outlier Factor):**
   LOF is an unsupervised anomaly detection method that computes the local density deviation of a given data point with respect to its neighbors [4]. Samples with substantially lower density than their neighbors are flagged as outliers.

4.2 Determining How the Data Was Manipulated

Once the manipulated cells are accurately identified, we will conduct a statistical analysis to reverse-engineer the manipulation process. This involves comparing the flagged values with the majority of data in the respective variables to assess their deviations and infer the modifications applied.

5 Final Results of Data Manipulation Detection

By the end of this project, we will have identified which methods are most effective for detecting manipulated data and will have evaluated the outcomes of the approaches described above. Our objective was to assess techniques capable of detecting anomalies, outliers, and manipulations in previously unseen datasets, mimicking real-world scenarios in which verifiers lack access to the true underlying data. These findings highlight the most promising tools for uncovering p-hacking and other forms of data manipulation.

References

[1] C. Raymaekers and P. Rousseeuw, *Challenges of Cellwise Outliers*, ScienceDirect, `https://www.sciencedirect.com/science/article/pii/S2452306224000078`.

[2] David K. Smith. *Cars Dataset*, Retrieved from: `https://rpubs.com/dksmith01/cars`

[3] scikit-learn, *Isolation Forest*, Available at: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html`.

[4] scikit-learn, *Local Outlier Factor (LOF)*, Available at: `https://scikit-learn.org/dev/auto_examples/neighbors/plot_lof_outlier_detection.html`.