Outlier detection for equipment data

Amine Laghaout

(Dated: September 21, 2016)

Executive summary: This report provides a brief synopsis of two data sets on which anomaly detection is performed. The data is assumed to be normally distributed and contaminated by a small proportion of outliers. No domain knowledge as to the variable space is assumed except that it is categorized by "equipment" type and logged as a time series. Further investigation is needed to determine automatically the proportion of contamination as well as to assert if there is a any equipment-dependence on the anomalies.

Keywords: outlier detection; anomaly detection; empirical covariance

Contents

I. THE PROBLEM

Two data sets `data1.csv` and `data2.csv` are provided with four features each: an integer equipment type `equipment`, a log date `date`, and two real, positive variables `variable1` and `variable2`. As will become evident later, the two variables `variable1` and `variable2`, though named identically in the two data files, cannot be assumed to be the same since `variable2` is seemingly discrete-like in `data2.csv` whereas it is continuous in `data1.csv`.

The unsupervised classification problem at hand is *outlier* detection, as opposed to novelty detection, since the data set is assumed to contain a finite, albeit small, proportion of outliers. The goal of this work is to devise a procedure for the identification of these outliers.

A. Assumptions

Three major assumptions shall be made. The first is that the number of outliers is small [5]. The second, which is based on the preliminary visualization of the data, is that the variables are linearly related. The third assumption, which seems better suited for `data1.csv` than for `data2.csv`, is that the data can be enveloped by a Gaussian distribution. The implications of this latter assumption will be discussed in further detail.

B. The outlier detection model

As mentioned above, the data is assumed to be drawn from a Gaussian distribution. By determining the mean and covariance of the data, this Gaussian distribution can be reconstructed. The heuristic we shall use to identify outliers thus consists of measuring the distance of the data points form the reconstructed distribution such that the furthest points are dismissed. In our case, the distance of a point from the presumed distribution is given by the Mahalanobis distance [1]. (The choice of the Mahalanobis metric is appropriate since it focuses on the distance between a point and a distribution, as opposed to other metrics—such as the Bhattacharyya or Hellinger similarity measures which are better suited for inter-distribution distances.)

II. DESCRIPTION AND SYNOPSIS

The purpose of this section is to visualize the data as well as the Gaussian envelope which presumably generates it. The data was first prepared by removing all the points at the origin as well as those that did not specify both variables. For each data set, we produce a scatter plot of the data, its inferred Gaussian envelope, and a linear fit. We also produce a scatter plot of the linear fits' slopes and intercepts for each equipment type to visualize the similarity in their behaviours.

A. Data set 1

Fig. 2 shows a scatter plot of the data. Little insight is provided by this plot since most of the data is confined close to the origin while the overall scale of the plot is biased by the outliers. One can expect that by removing

	`equipment`	`variable1`	`variable2`
mean	34.9	8.2	141.3
std. dev.	24.4	26.0	524.5
min	0	0.0	0.0
50%	34	7.5	116
max	92	965.4	25779.5

FIG. 1: Synopsis of the data in `data1.csv`: A quick comparison of the maximum values of the variables with their standard deviations can be used as evidence for outliers. For example, the maximum of `variable2` is 25779.5 whereas its standard deviation is 524.5. Given the Gaussian nature of the distribution, the maximum value can at once be declared as an outlier.

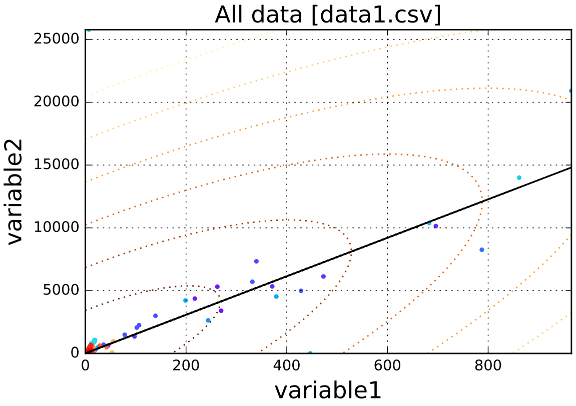

FIG. 2: All the data in `data1.csv` is plotted in this figure along with its inferred Gaussian envelope and linear fit. Each colour corresponds to an equipment type. (Since we are dealing with 92 equipment types, the legend is omitted so as to avoid clutter of the figure.)

the outliers, the relevant pattern will be better visible. This is done in Sec. III A.

Fig. 3 does not directly relate to anomaly detection *per se*, but instead helps determine whether the different equipment types are similar enough to justify bundling them together under the same Gaussian envelope. (By visualy inspection, this seems to be the case except for about five equipment types.)

B. Data set 2

This subsection repeats the analysis above for the data in `dat2.csv`. See Table 4 as well as Figures 5 and 6.

III. OUTLIER REMOVAL

In this section, we shall identify the outliers and reproduce the scatter plots with the inliers only so as to recover the "true" patterns of the data. In order to do so, we define a proportion `contamination` $\in [0, \frac{1}{2}]$ of outliers which shall be set manually.

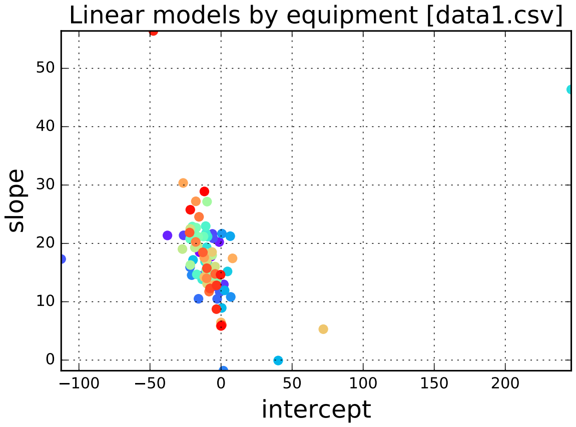

FIG. 3: The data for each equipment type is subjected to a linear fit and the slopes and intercepts are then plotted so as to reveal the similarity in their linear models.

	`equipment`	`variable1`	`variable2`
mean	34.2	240.8	3.2
std. dev.	24.3	167.9	1.9
min	0	0.0	0
50%	33	215.1	3
max	92	1371.2	11.36

FIG. 4: Synopsis of the data in `data2.csv`: A quick comparison of the maximum values of the variables with their standard deviations can be used as evidence for outliers.

A. Data set 1

From Fig. 7 can be seen that the distribution of distances is smooth up to a natural kink. This natural feature in the data seems to be a good candidate for the cutoff between inliers and outliers. In this particular case, it corresponds to a proportion of outliers

$$\texttt{contamination} = \frac{1}{17}. \qquad (1)$$

B. Data set 2

The single kink that could easily be seen in `data1.csv` is now absent. We have therefore chosen an arbitrary cutoff at

$$\texttt{contamination} = \frac{1}{10}. \qquad (2)$$

Instead, several other kinks appear along the sorted distribution of distances which may arise from equipment-dependence. In other words, the different equipment

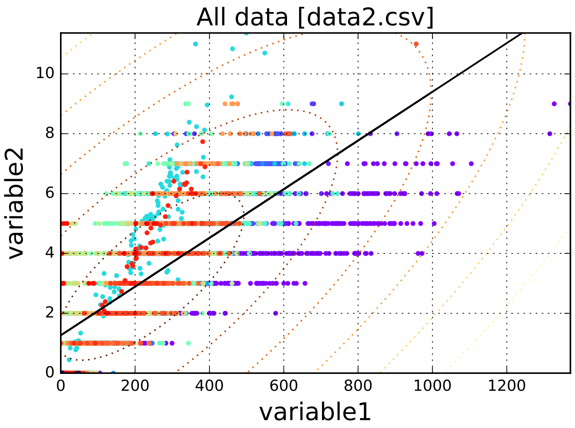

FIG. 5: All the data in `data2.csv` is plotted in this figure along with its inferred Gaussian envelope and linear fit. Each colour corresponds to an equipment type. (Since we are dealing with 92 equipment types, the legend is omitted so as to avoid clutter of the figure.)

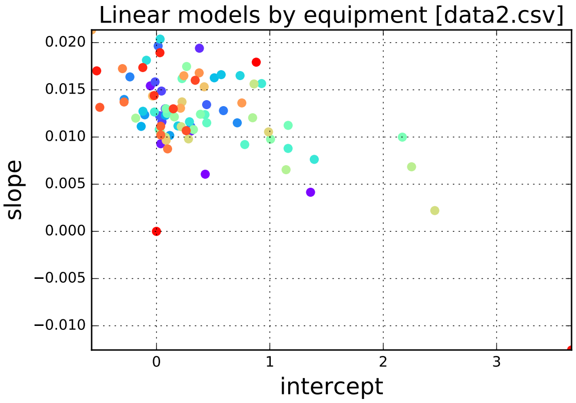

FIG. 6: The data for each equipment type is subjected to a linear fit and the slopes and intercepts are then plotted so as to reveal the similarity in their linear models.

types may not be justifiably bundled by the same model. (This is incidentally hinted at by the more spread-out scatter in Fig. 6 than for `data1.csv`.)

IV. CONCLUSION AND OUTLOOK

We have developed a method to identify outliers in the data based on the assumption that it can be enveloped by a Gaussian distribution. The results are more conclusive for `data1.csv` than for `data2.csv` since all equipment types can be seamlessly bundled into a single envelope in the former but much less so in the latter. In view of this, the analysis of outliers could be made more accurate if performed separately for each individual category.

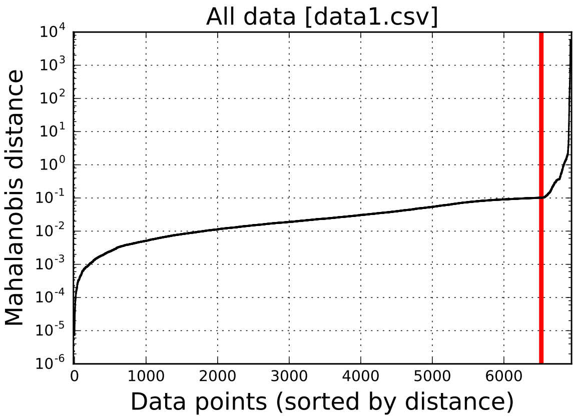

FIG. 7: The data points are aligned in the order of increasing Mahalanobis distance to the Gaussian envelope. The red line represents the threshold beyond which the data is assumed to be made up of outliers.

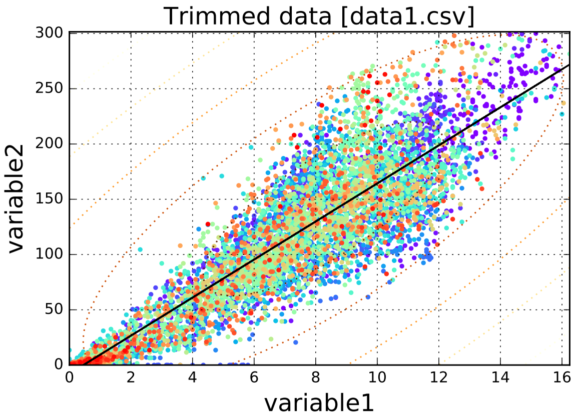

FIG. 8: The scatter plot of the data in `data1.csv`, after pruning out the outliers.

A. Sensitivity to outliers

The Gaussian envelope was determined with the Maximum Likelihood method using the Python function `EmpiricalCovariance`. The shortcoming of this method is that it is sensitive to outliers. In other words, outliers introduce a bias in the reconstructed Gaussian which we hope to be small—but cannot guarantee to be so—by assuming that the contamination ratio is small. This problem was address by Rousseeuw in [2]: Different subsets of the data can be probed to identify the one which best corresponds to a Gaussian distribution. That subset will then be used as the basis of inliers [3]. A method already exist in Python which implement Rousseeuw's technique of Minimum Covariance Determinant, namely `EllipticEnvelope`. However, it turns out that this method is prone to numerical instabilities and we instead opt for the simpler—yet more naive—method of `EmpiricalCovariance`.

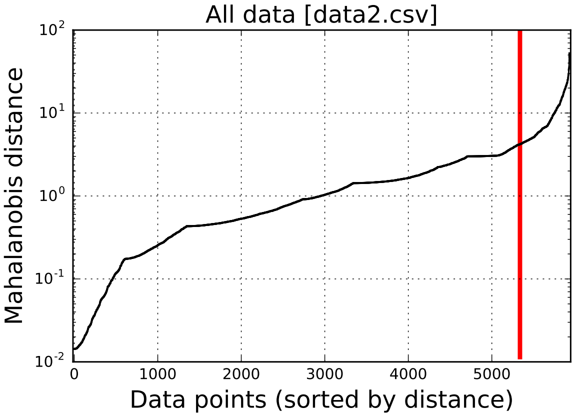

FIG. 9: The data points are aligned in the order of increasing Mahalanobis distance to the Gaussian envelope. The red line represents the threshold beyond which the data is assumed to be made up of outliers.

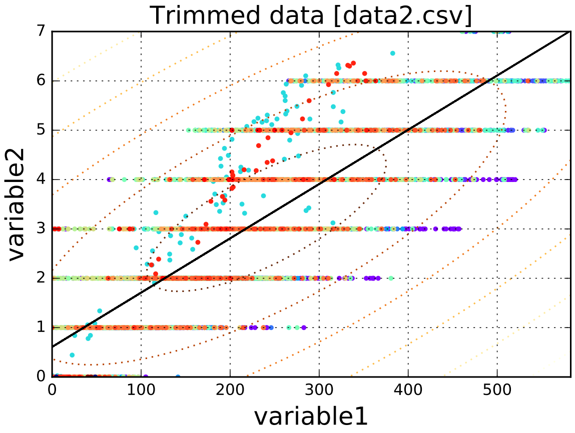

FIG. 10: The scatter plot of the data in `data2.csv`, after pruning out the outliers.

B. Adaptation to non-Gaussian distributions

Although the assumption of Gaussianity seems justified for `data1.csv`, it is less so for `data2.csv`. In this case, a One-Class SVM classifier may be better suited to produce an envelope of the data for each category [4].

C. Determination of the contamination ratio

The kink observed in 7 offered a compelling indication for choosing the contamination ratio. However, a more systematic method, based on second-derivatives may allow us to detect the cutoff automatically.

D. Further considerations

In addition to the points mentioned above, further investigation is called for, namely:

1. Break-down the analysis by type of equipment

2. Merge the data files based on equipment type *and* date. The analysis shall then be repeated in the four dimensional space.

3. Time-dependence should be investigated: Are there days with more anomalies, perhaps due to a temporal trend or temporary bias/breakdown in the data acquisition process?

[1] P. C. Mahalanobis, in *Proceedings National Institute of Science, India* (1936), vol. 2, pp. 49–55, URL `http://ir.isical.ac.in/dspace/handle/1/1268`.

[2] P. J. Rousseeuw, Journal of the American Statistical Association **79**, 871 (1984).

[3] Scikit-learn, *Covariance estimation with scikit-learn*, URL `http://scikit-learn.org/stable/modules/covariance.html`.

[4] Scikit-learn, *Outlier detection with several methods*, URL `http://scikit-learn.org/stable/auto_examples/covariance/plot_outlier_detection.html`.

[5] The exact proportion of outliers can either be declared arbitrarily—as is the case here—or elicited from the data.