Progression overview

April 20, 2018

This is a progression overview of Outlier Factors for Device Profiling. All coding examples can be found at Github (not very tidy an the moment).

We will focus with the following dataset Los Alamos National Laboratorys corporate, internal computer network. A artificial generated dataset could be used as well, but the nature of the data should be closely taken into account.

1 Distribution

The data used comprises of a collection of flows created at different timestamps from different users. In a first attempt we will only consider the number of flows and the bytes sent through these flows. We will also temporarily ignore time relations and drifts in the data.

The beginning of this dataset can be displayed below:

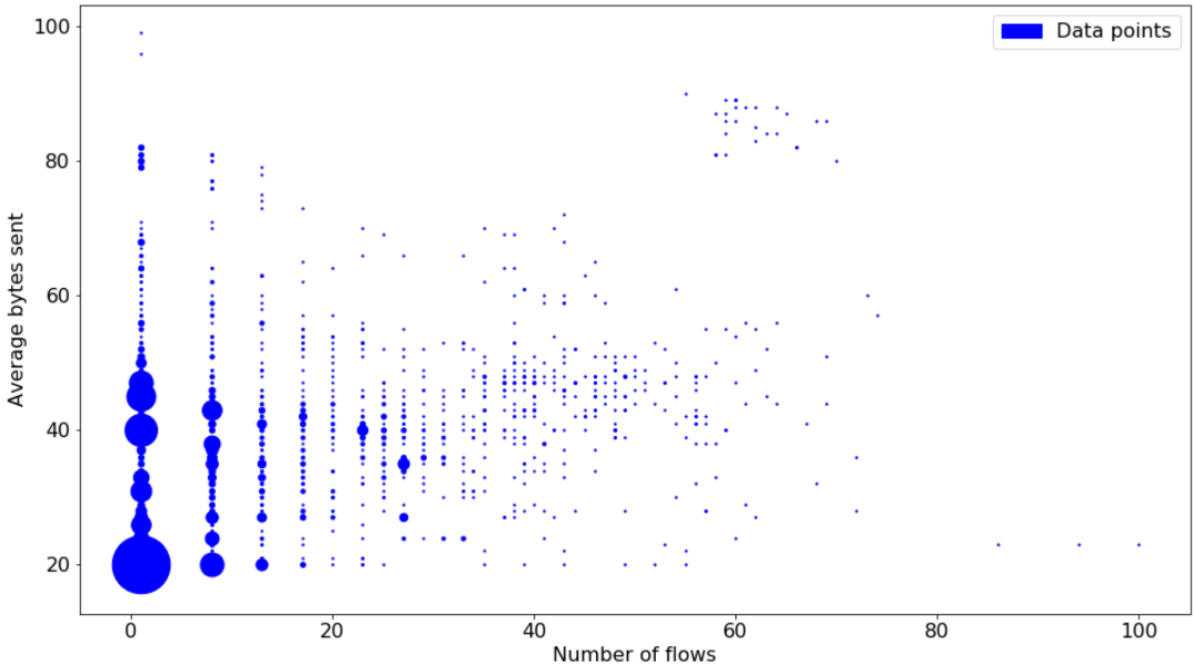

Figure 1: Distribution (the size of each dot denotes the number of points with the same value)

The data have been transformed with the function log(x + 1) and scaled in the range of 1 - 100 as a type of int (required for the Poisson distribution). We have also considered that the two dimensions (count of flows and average bytes sent) are independent from each other.

The measure of average bytes and number of flows was achieved after bucketing the data in time periods defined by the user.

- Issue 1

  This time period may affect the responsive time of the algorithm.

The approximating algorithm for the Poisson distribution is a multidimensional version of the algorithm found OnlineEM.

There are a couple of problems with this approach:

- **Issue 2**

  Not all points can be formally approximated, as an underline Poisson distribution may be arbitrary and biased. In the following figure these data points have been approximated with a 20-mixture Poisson.

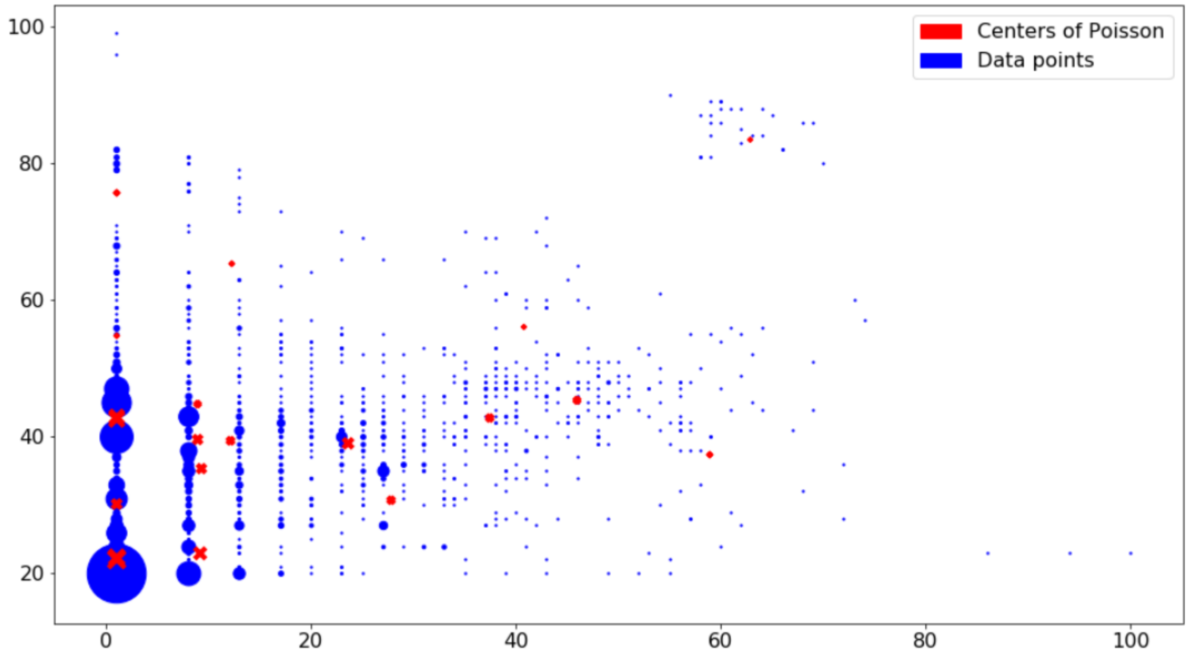

Figure 2: Poisson center for this distribution. The size of the red x-es denotes the probability that a random next point will belong to the current distribution (referred as gammas in the paper).

  Not all points can be represented adequately through this mixture (probably points not anomalous too). Also due to the nature of the data, some of the mixture centers tend to converge.

- Issue 3

  As proposed by the paper, in order to guarantee convergence of the algorithm, the update factor in its iteration should satisfy the following:

$$a_n > 0, \quad a_n \to 0, \quad \sum_{n=1}^{\infty} a_n = \infty, \quad \sum_{n=1}^{\infty} a_n^{1+\delta} < \infty,$$

  This could later be an issue as our algorithm will adapt increasingly slowly with changes. This problem can be faced in later stages (through some kind of stability coefficient perhaps).

2 Clustering

Our purpose is to cluster the behavior of the users in groups with similar behavior. Each group will allow the appearance of data points in certain mixtures (we use the term mixtures to denote the mixtures of Poisson distributions to differentiate from the clusters of different Poisson created at this step).

As a first naive implementation, the following is executed.

Each host calculates the mixtures he is mostly related with, by averaging the mixtures each of his data points is related to

- Issue 4

  However these mixtures (their parameters) will change over time so this method cannot adapt. Perhaps use exponential forgetting.

These relations with each mixture vectors are then clustered with a regular clustering algorithm. (As these relation-vectors are evolving through time the algorithm should somehow adapt also).

We have used 8 clusters to represent 8 different acceptable behavior patterns from the users. This may appear to be too much, but in other behavior grouping such as work profiling in data centers a lot more may be used (example)

These clusters can be seen in figure 3.

3 Flag anomalies

For flagging anomalies a weighted approach should be used between the past history of each host's activity and the group he belongs to. The history basically comprises of an array of size number_of_mixtures, showing the relationship of the history with each Poisson distribution.

We are taking into account that the number of anomalies will be significantly lower than the number of normal data points. So no mixture will be affiliated closely with anomaly points. As a result, a user with anomalous behavior in the past will not be flagged as normal due to similar past behavior. On the contrary, his past will be related with the closest mixture center, which should be relatively far away.

The following results are taken averaging the anomaly score from the group and the individual history. An anomaly is flagged if outside a threshold from the cumulative distribution function.

- Issue 5

  How to flag in a multidimensional scene? We could just multiply the results and the thresholds.

We tested the first 20,000 data points indicating 20,000 flows. A bucketing of every 50 seconds was made leading to 5381 individual data points from 1576 different users. With a threshold of 0.01, 15 anomalies where indicated from 11 different used. Below some of these anomalies are shown:

An API could be used for an administrator to indicate if an anomaly actually occurred or not. According to this future anomaly score should be adapted (per host and in general).

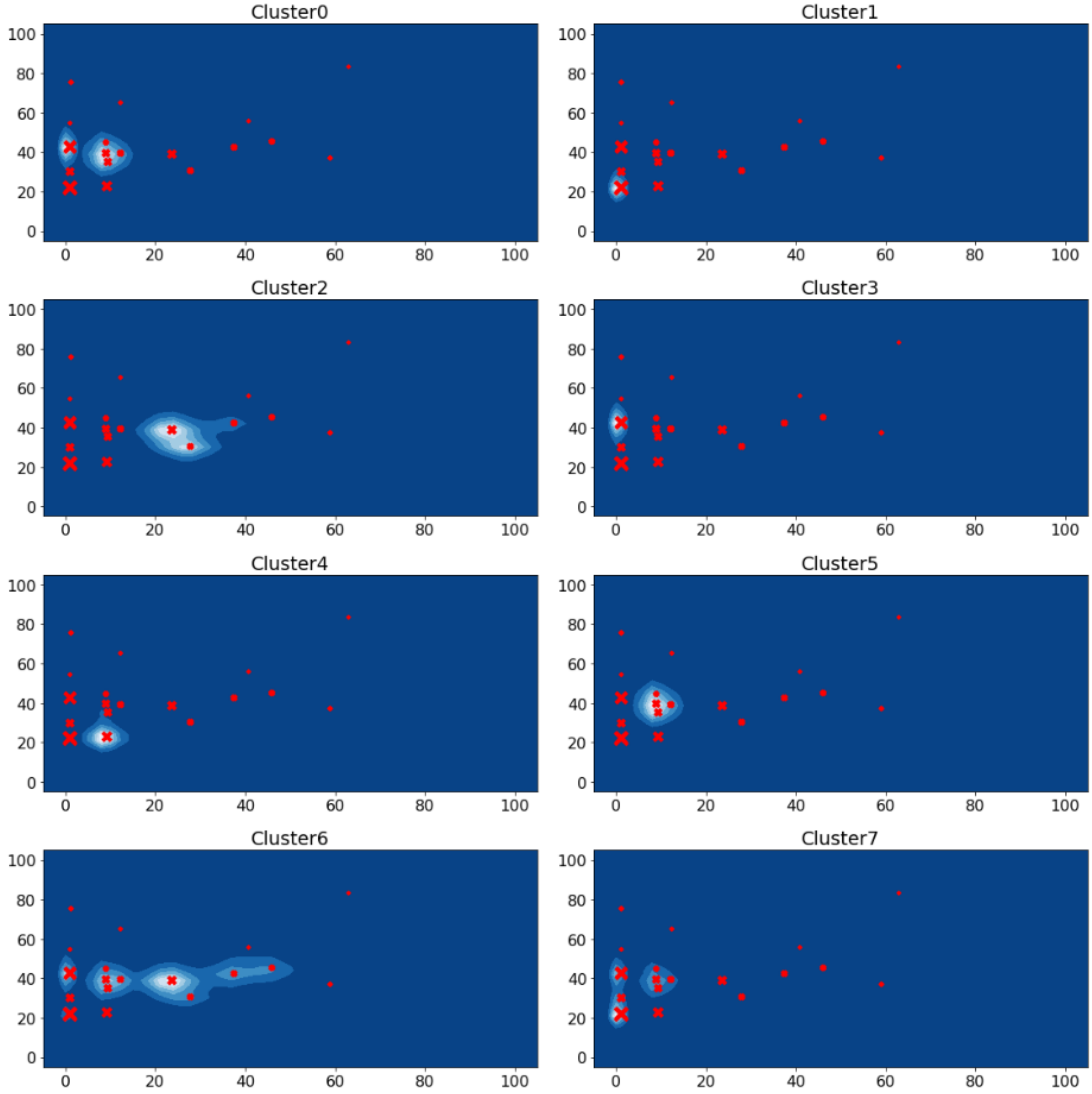

Figure 3: Clusters created for different groups of behavior

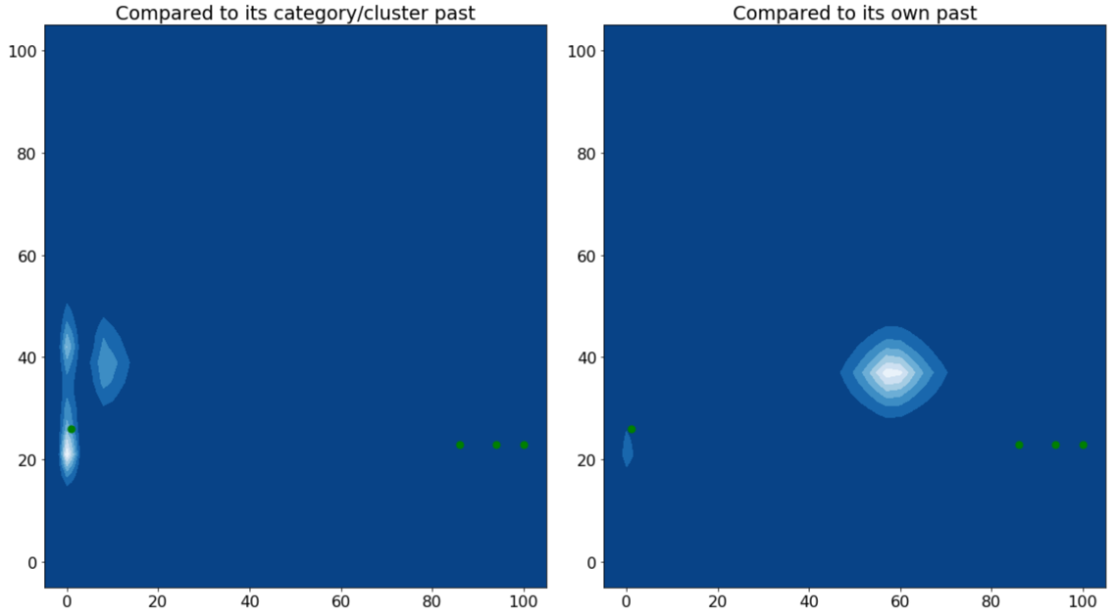

Figure 4: User C16712 has 3 anomalous data points. The cluster he is attributed from is very far away from his data points. Also the closest mixture as shown in the right figure is still far away from his behavior. This could indicate anomalous behavior.

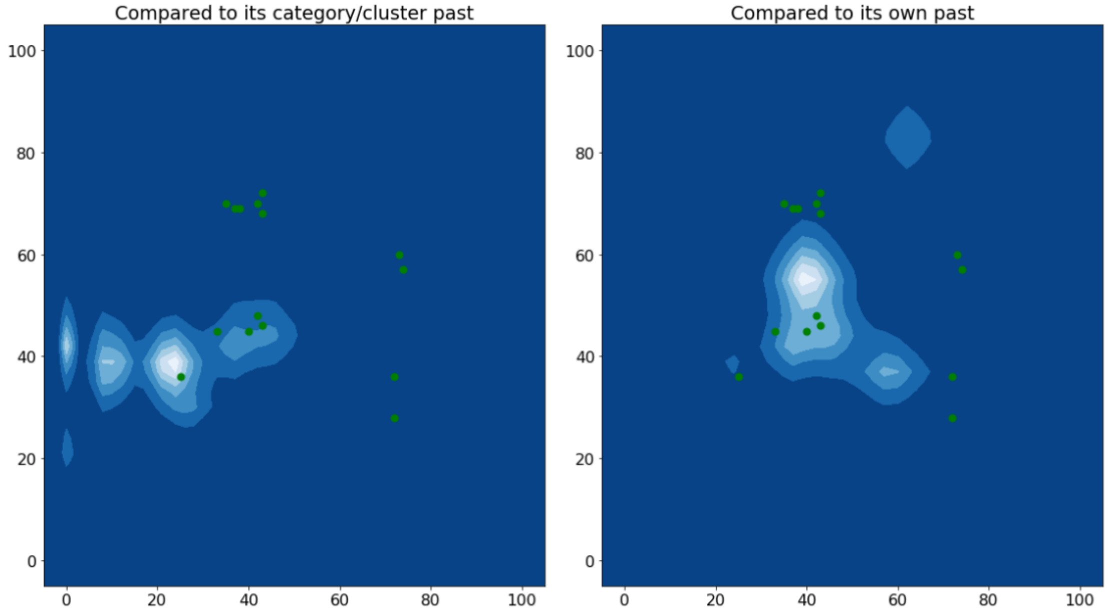

Figure 5: User C16712 has 2 anomalous data points. Similarly the closest mixtures are far away. The number of points is also low to indicate a clear behavior.

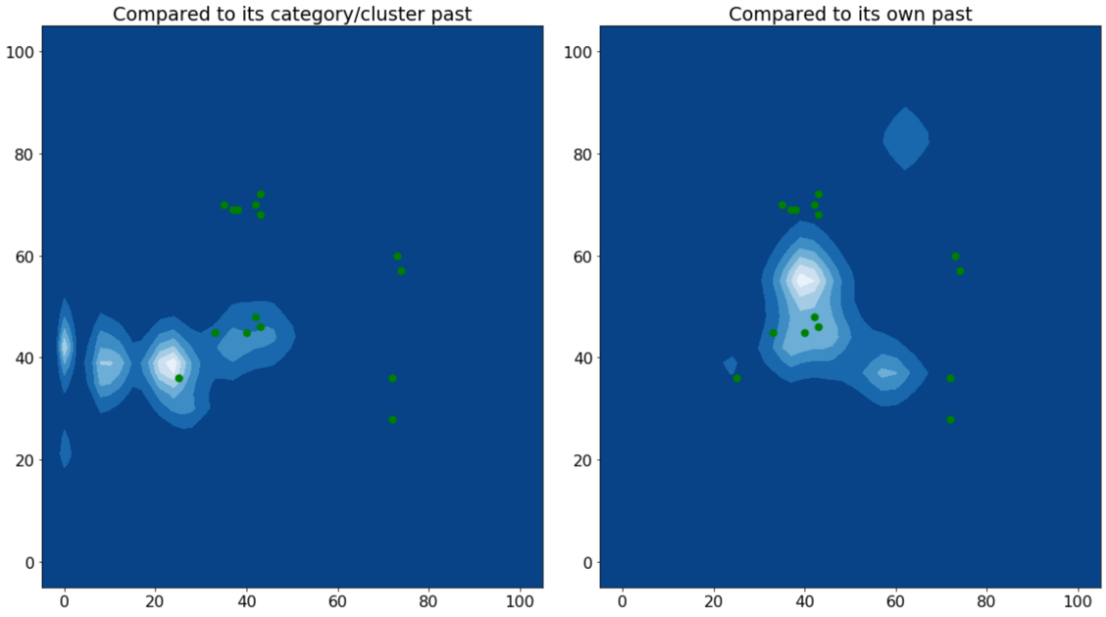

Figure 6: User C17926 has 2 anomalous data points.

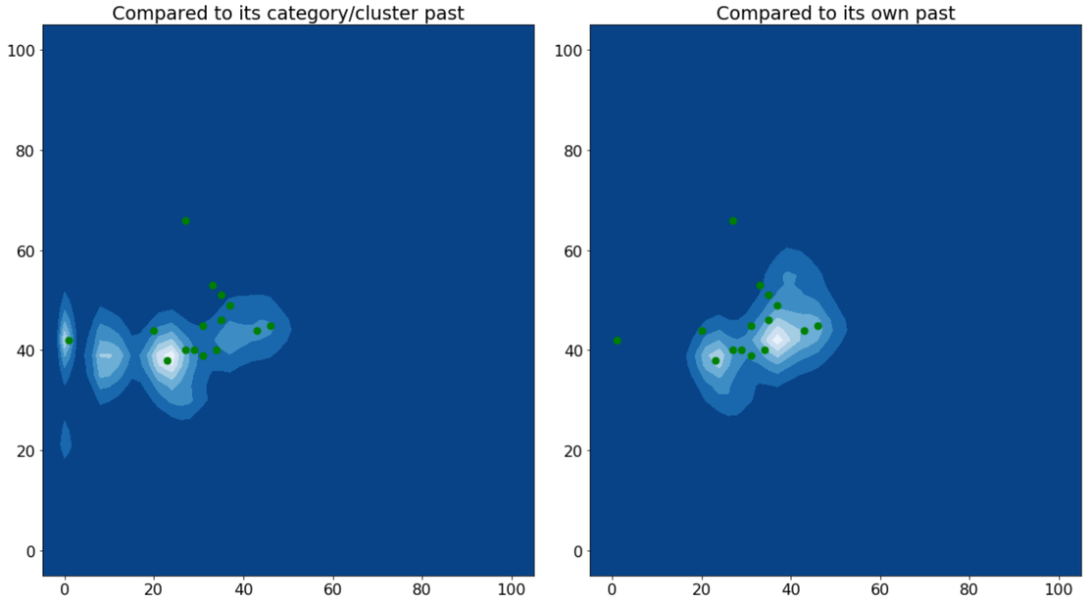

Figure 7: User C625 has 1 anomalous data point.