Progression overview

May 10, 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 in 1. As we can see most points denote a zero traffic for users in a given time.

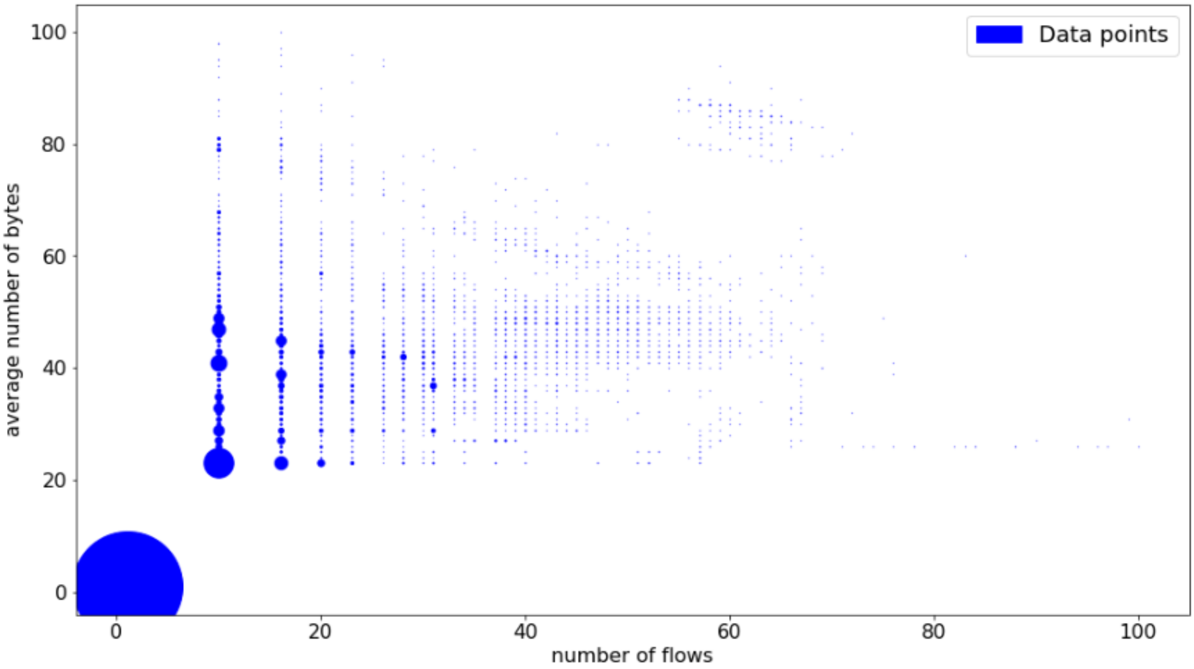

Figure 1: Distribution (the size of each dot denotes the number of points with the same value)

The main interest is how the behavior of a host at a given time (epoch) compare to the host past behavior and the time field (epoch).

The fact that most of the hosts have a zero traffic most of the time can be illustrated in 2, where we can see the average traffic for each individual host.

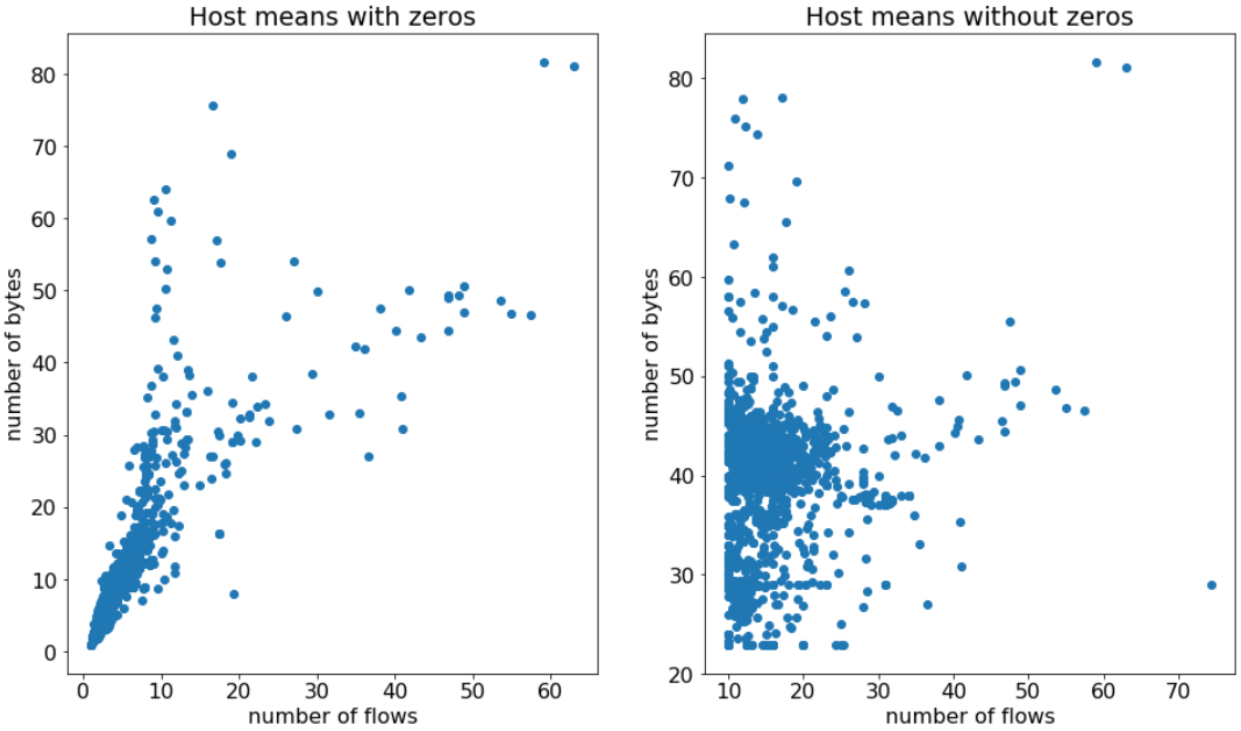

Figure 2: Average traffic for each individual host.

In the figure 3 we can see the average traffic for each individual epoch.

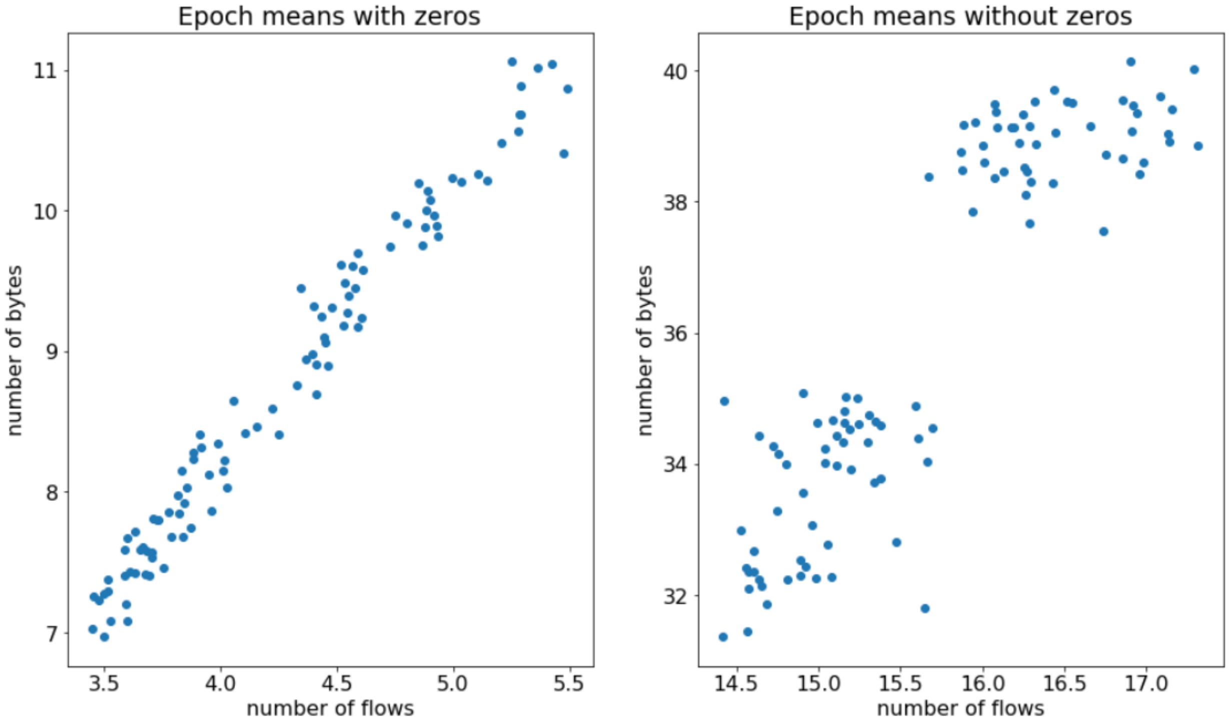

Figure 3: Average traffic for each epoch.

	Mean Square Error
Each point compared to the mean of the host's past	201.481193
New mean for host compared to the mean of the host's past	3.26980036

Table 1: Mean Square error comparing to each host individually

	Percentage of non zero traffic	Mean Square Error
Individual points	0	201.481193
	10	231.04849494
	20	242.53839645
	30	216.13443677
	40	240.51540553
	50	228.43204413
	60	213.30825695
	70	187.704809320
	80	153.79107752
	90	99.99244057
Average of points	0	5.57316684
	10	4.2962690363
	20	4.3410310789
	30	4.387796819
	40	2.1611752767
	50	1.9575657276
	60	2.4987264900
	70	2.9684499999
	80	2.427106741
	90	2.4334405797

Table 2: Mean Square error comparing to each host individually depending on the host's traffic.

To evaluate these methods we will use as a test a new part of the dataset. Firstly we compare new traffic to the host's history. The results can be seen in table 1. We can make some useful observations:

- When we compare each data point separately to the host's past, the mean square error is much higher. This further enforces our point for a mixture of models or clusters.
- The average traffic of each host is relatively stable as shown by the error value.

We will further investigate these facts by examining hosts that experience a higher percentage of traffic (that is less periods with zero traffic). The results can be seen in table 2. As the number of non zero traffic rises, we can see a small increase in the MSE in the case we investigate each individual data point separately. This should probably be expected. After a point the MSE decreases. This is due to the fact that many hosts with high traffic, have a very stable kind of traffic. This perhaps may due to the fact that certain network devices have a predefined role (e.g. DNS resolvers).

Next we will compare relatively to the epochs past. This method obviously has major drawbacks, as comparing each individual data point will have as a result a huge loss. An epoch is an average of many, different between them, events. The individual MSE within the same epoch actually is 297.68073.

Symbol	Description
L	The number of features (2 in our case)
N	The number of hosts
M	the number of lambda mixtures
K	The number of k-means clusters
h_i	Host i
e_j	Epoch j
l_k	Lambda parameters of cluster k (vector size of L)
g_k	Gamma paramters of cluster k
c_k	Kmeans cluster k

Table 3: Major notations for online EM alogrithm.

2 Online EM

Before comparing different methods, we should apply some notations as shown in table 3

Next we will how the onlineEM algorithm can cope with the same data points. In the figure 4, we can see the parameters of the model after the first fit. It is difficult to compare this method to the previous one, so we will just compute the log-likelihood in the case clusters are taken into account or not. The results can be seen in table 4.

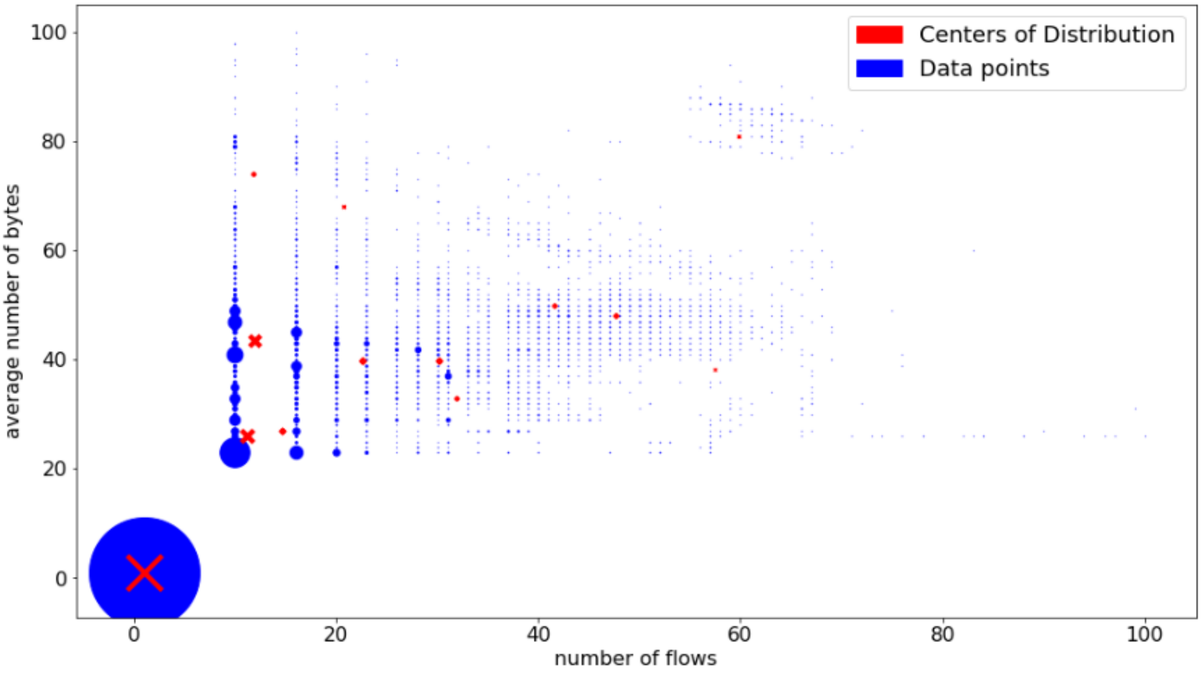

Figure 4: Data points.

2.1 Heatmaps

The figures 5 and reffig:heatmap-epochs display the heatmaps for the hosts and epochs across all clusters.

The table 5 shows what the cluster parameters are

Per of non zero traffic	no clusters	clusters	per diff	hard hmm	per diff	soft hmm	per diff
0	-3.62652	-3.38854	-6.56225	-3.54289	-2.30609	-3.54456	-2.26018
10	-4.07975	-3.77126	-7.56151	-3.96889	-2.71751	-3.97039	-2.68061
20	-5.17374	-4.60974	-10.9012	-4.95994	-4.13248	-4.96237	-4.08553
30	-5.50981	-4.81983	-12.5228	-5.24119	-4.87523	-5.24477	-4.81038
40	-6.69932	-5.61709	-16.1542	-6.17865	-7.77195	-6.18003	-7.75132
50	-7.23418	-5.91754	-18.2001	-6.55856	-9.33928	-6.56159	-9.29733
60	-8.01801	-6.31685	-21.2167	-7.05459	-12.0156	-7.06148	-11.9297
70	-8.61507	-6.57927	-23.6306	-7.39089	-14.2097	-7.39826	-14.1241
80	-9.18878	-6.83602	-25.6046	-7.64999	-16.7463	-7.65902	-16.6481
90	-9.56419	-7.02990	-26.4976	-7.79269	-18.5222	-7.79660	-18.4813

Table 4: Log-likelihood depending on the amount of traffic from each host.

w

Cluster number	number_of_flows	average_number_of_bytes	gamma
0	1.0	1.0	0.772517
1	59.8762067715	80.822602599	0.0009252
2	11.2184578545	25.940333816	0.0963219
3	11.7888754317	73.976797269	0.0033327
4	31.9397765315	32.835940994	0.0034302
5	22.6224235462	39.821418419	0.0129561
6	11.9241334516	43.583919445	0.0765875
7	57.5227356986	38.164442817	0.0006072
8	47.6478220471	47.984080424	0.0063930
9	14.6902086613	26.959186862	0.0109853
10	20.6797204108	67.995234687	0.0010794
11	30.2154115845	39.731843682	0.0112914
12	41.5515493950	49.858997457	0.0035721

Table 5: The parameters of the em algorithm.

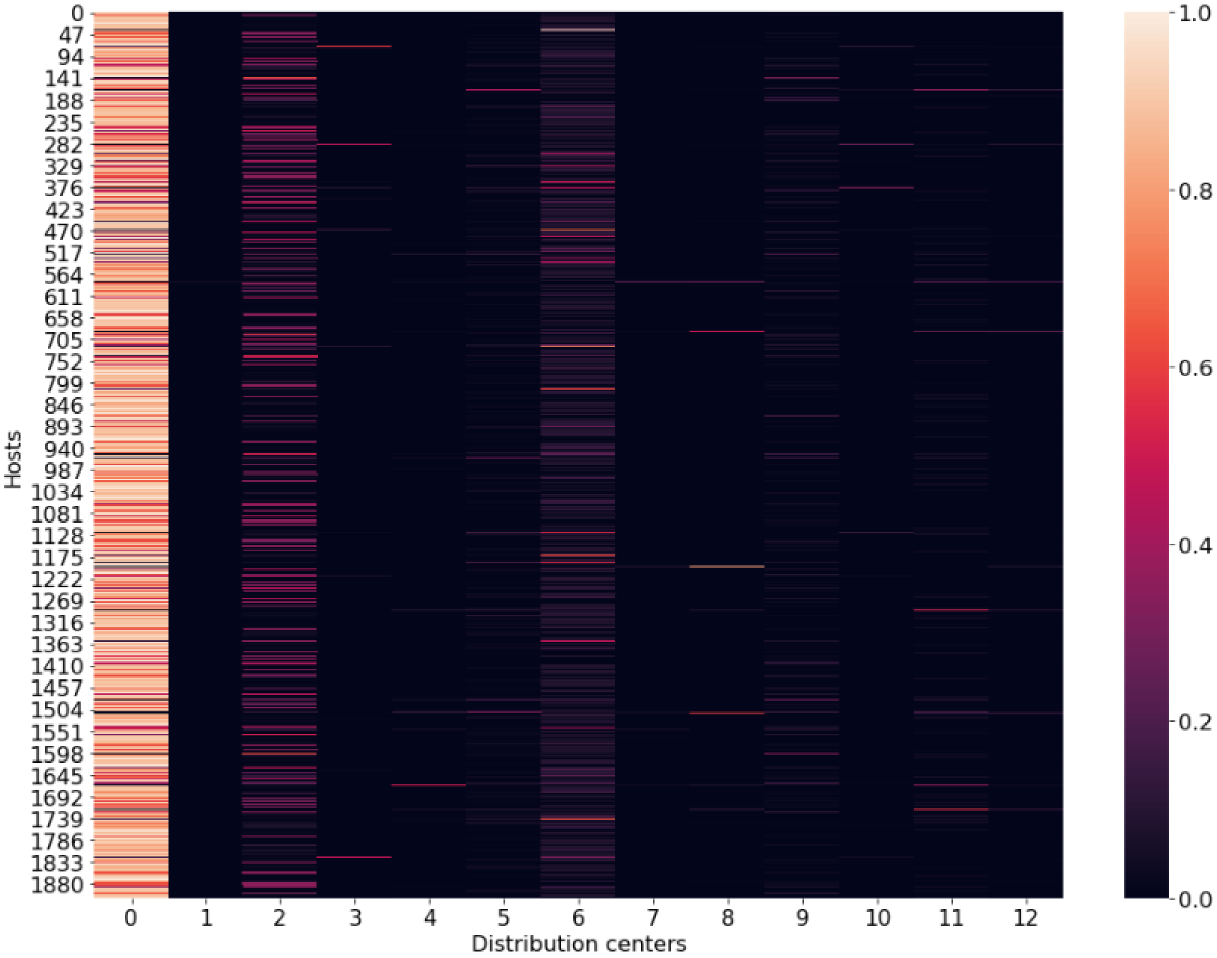

Figure 5: Heatmap for hosts.

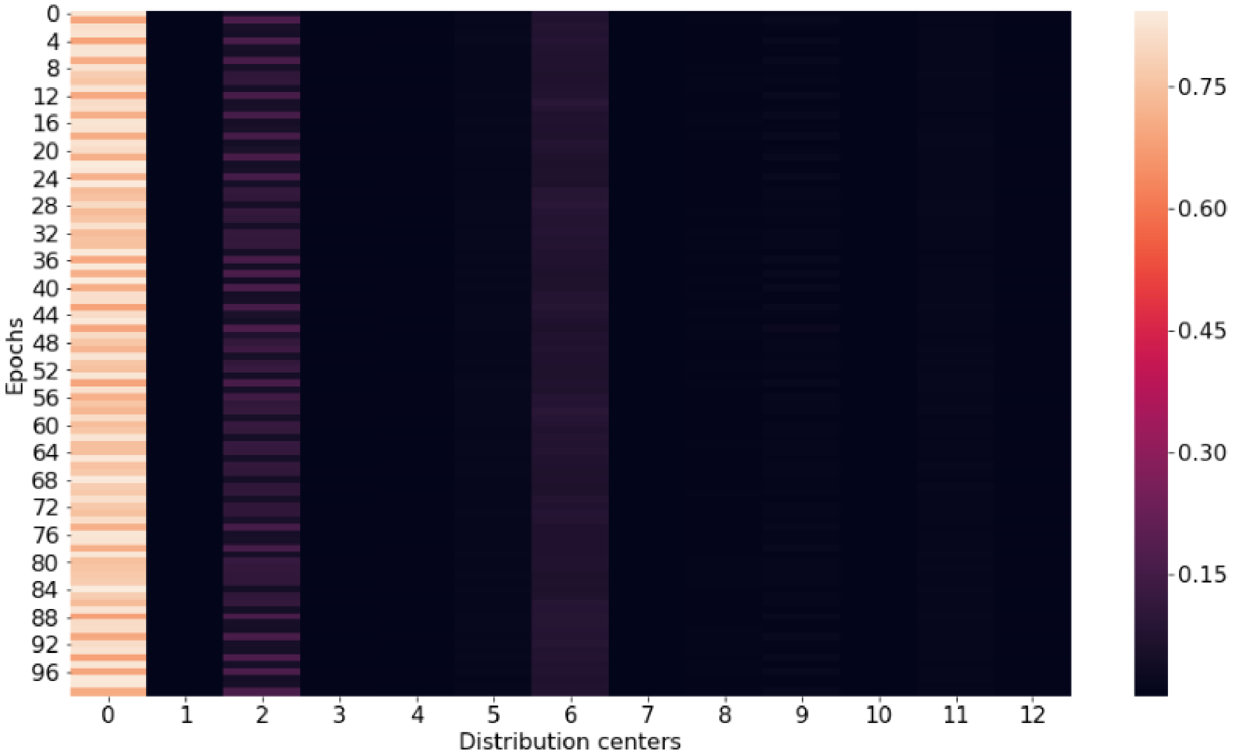

Figure 6: Heatmap for epochs.

3 Transitions between clusters

Finally we compute the transition matrix for every cluster. This can be computed either using hard or soft clustering.

- Hard clustering: For each new data point arriving, we calculate the likelihood it belongs to each of the M lambda centers. The center with the highest likelihood value is denoted to be the new cluster to which the host is acquainted with. We create the transition matrix which is a MxM matrix where each individual points t_{ij} shows the probability that with a previous state a i, the next state will be j.

  In this case, to calculate the log likelihood we replace the gammas parameters, which show the probability of each individual cluster, with the transition matrix probabilities, based on the previous state for each individual host.

- Soft clustering: In this case the previous state is not implicitly known. Instead a vector of size M in known, containing the probabilities the previous state was part of the one of the M distributions. By this we update the transition matrix of each of the M clusters by a factor equivalent to the probabilities of the previous state.

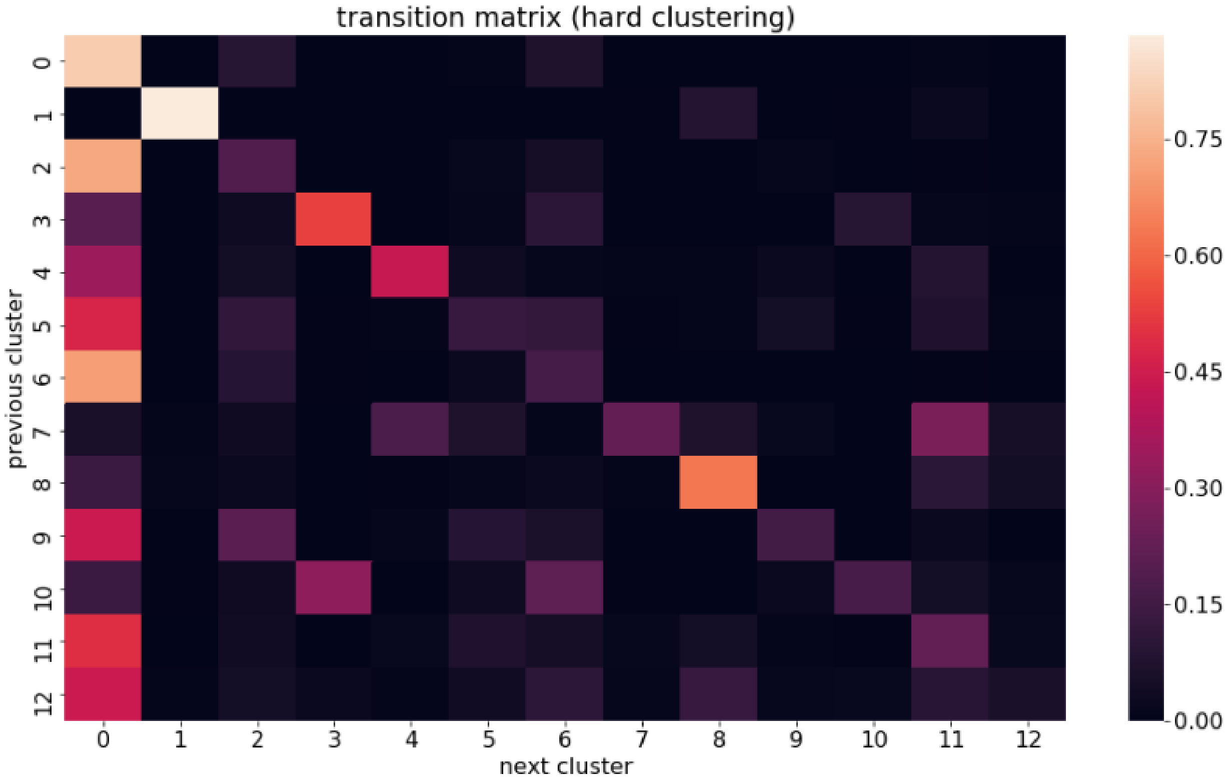

Figure 7: Heatmap for transitions for the em clusters (hard clustering).

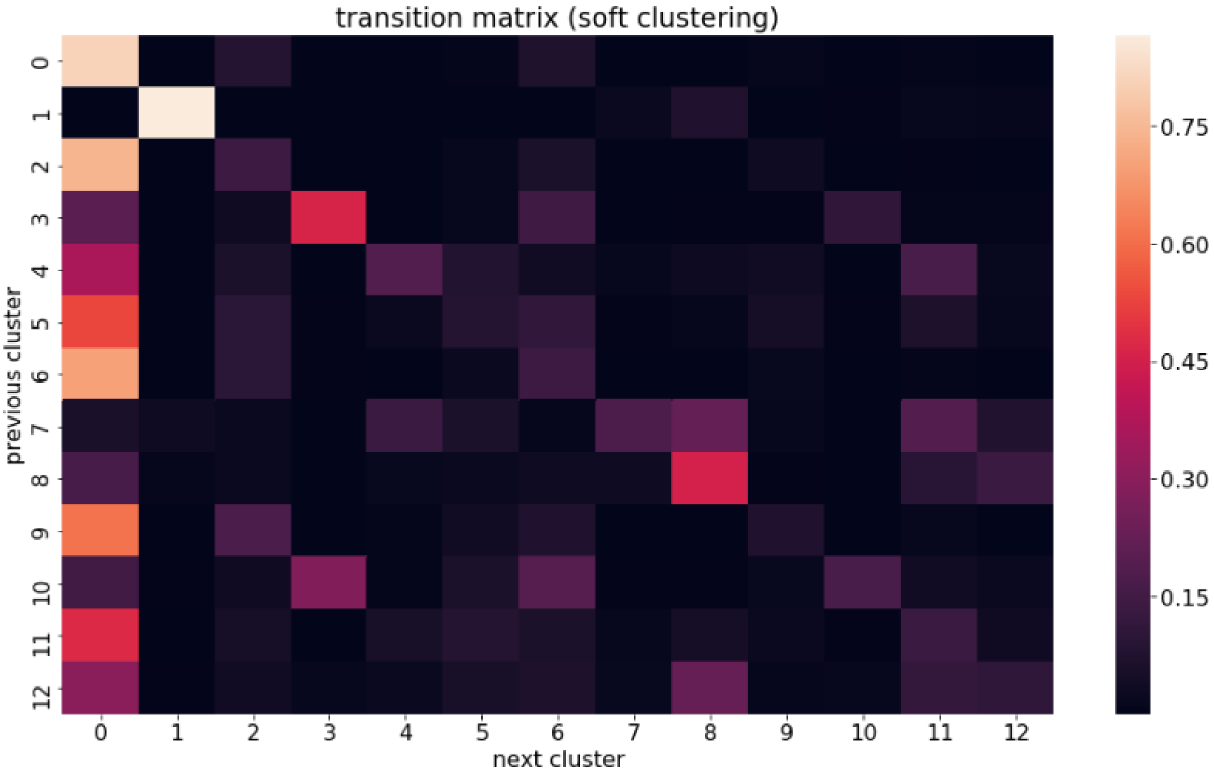

Figure 8: Heatmap for transitions for the em clusters (soft clustering).