Dirichlet Process Mixture Models
Deliverable 3 - Evaluation of the Model

A. Mardale, M. Poul Doust, L. Couge, E. Ekpo

January 28, 2020

1. **What did the model learn?** The different plots produced below are produced using TSNE. It is to be noticed that the location of the data point may not contain relevant information. We are more interested in the coloring of the points.

   (a) Run on training data

   The algorithm forms 8 clusters (Figure 1). The visual inspection of the time series and frequency spectra does not allow finding any pattern in the clustering. We don't have enough information on the sequences.

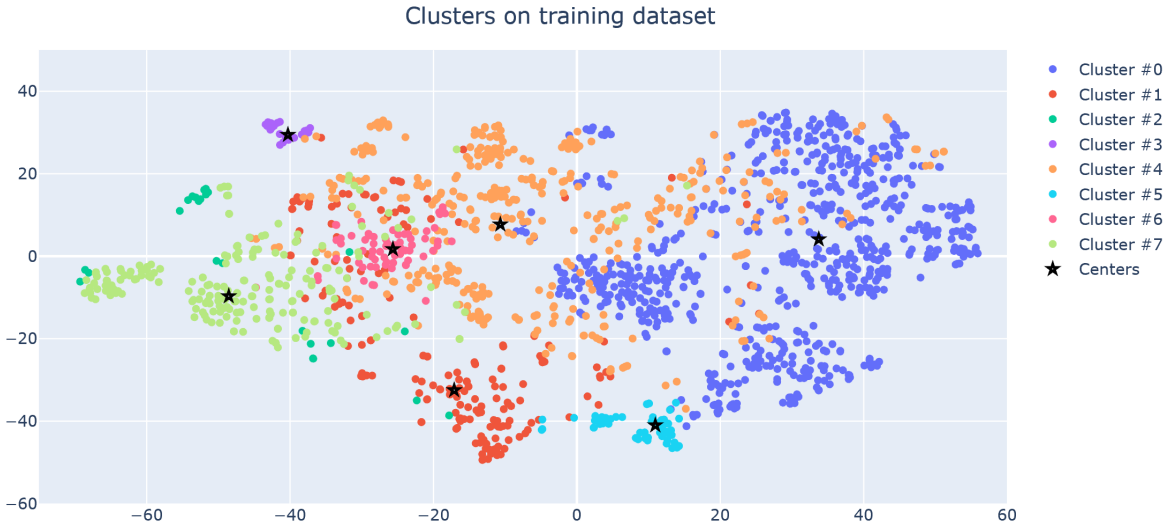

Figure 1: Clustering on training set

   (b) Run on validation data

   We get 5 clusters on the validation set (Figure 2). When we superpose the clustering to the anomalies, we find out that clusters 0, 1, 3, 4 contain only anomalies, cluster 2 contains mainly negative samples but also few anomalies. The algorithm is able to cluster most of the anomalies without using the training set.

Figure 2: Top: Clustering on validation set - Middle: anomalies (in red) on validation set using same coordinates as above - Bottom: anomalies (in red) on the cluster 2

(c) Run on the 3 datasets (training, validation, testing)

We get 10 clusters using the 3 datasets (Figure 3 top). When we compare with the ownership of the points in the 3 datasets (Figure 3), we observe that the clusters of the training and the testing are completely separated. Some clusters mixed data points from validation and testing. The testing dataset contains about 77% of negative samples. Those samples have a distribution completely different from the training set, so we can wonder if it was really possible to detect anomalies using more classic machine learning approaches, when using training set only.

Figure 3: Top: Clustering on 3 datasets: training, validation and testing - Bottom: Identification of the 3 datasets with same coordinates as above

2. **Is the model suitable for the data ?**

Our model is not directly suitable for the provided raw time series dataset due to the data high feature dimensionality. Indeed, each sequence within our training and validation dataset is composed of $\sim$ 60000 data points/features. This high feature dimensionality led us to apply some features extraction techniques in order to learn a new lower representation of 120 data points from the

original data. While some relevant information may have been lost during the encoding step, the new representation learnt prevented our model from suffering the curse of dimensionality, thus less prone to overfit.

3. **Is the inference working?**

   We believe that the inference is working fine. There are two aspects which lead to this conclusion: firstly, by taking into account the results we obtain on the Airbus dataset, especially on validation and on test set, that seem to be really meaningful. Secondly, we correlated the effect of the priors on the results (eg: number of obtained clusters) and observe that they confirm the mathematics intuitions behind this model.

   To conclude, the Collapsed Gibbs Sampling inference method is working fine as long as we are using conjugate priors.

4. **Is it sensitive to initialization, to hyperparameters, to the quality of the priors?** The applied algorithm (Dirichlet Process Mixture) using collapsed Gibbs sampling seems to be highly sensitive to different initialization and priors where each prior impact directly the final results. Moreover,

   (a) **Priors:** In the proposed model, we have priors for the Gaussian-Inverse-Wishart prior distribution on μ and σ of a multivariate Gaussian distribution. Specifically, **m0** is the prior mean for mean μ, **k0** indicates how much we are trusting m0, **S0** proportional to the prior mean of Σ and **vo** represents the confidence in S0. Additionally, the model is parameterized by α (concentration parameter) which affect the number of clusters. Figure 5 depicts the effect of the hyper-parameters on the final results.

   (b) **Initial Assignments**: The method chosen to initial the assignments highly impact the clustering algorithm. As an initialization method several methods could be applied. For example, consider each point as a separate cluster or random assignments vector.

   (c) **Number of iterations**: As the algorithm is an iterative one, the number of iterations is of an important role for the quality of the results. For example, we can see from Figures. 6, 7 how the same problem with different number of iterations affected the final number of the cluster in addition to the quality of each cluster (few iterations results in non-intuitive clusters).

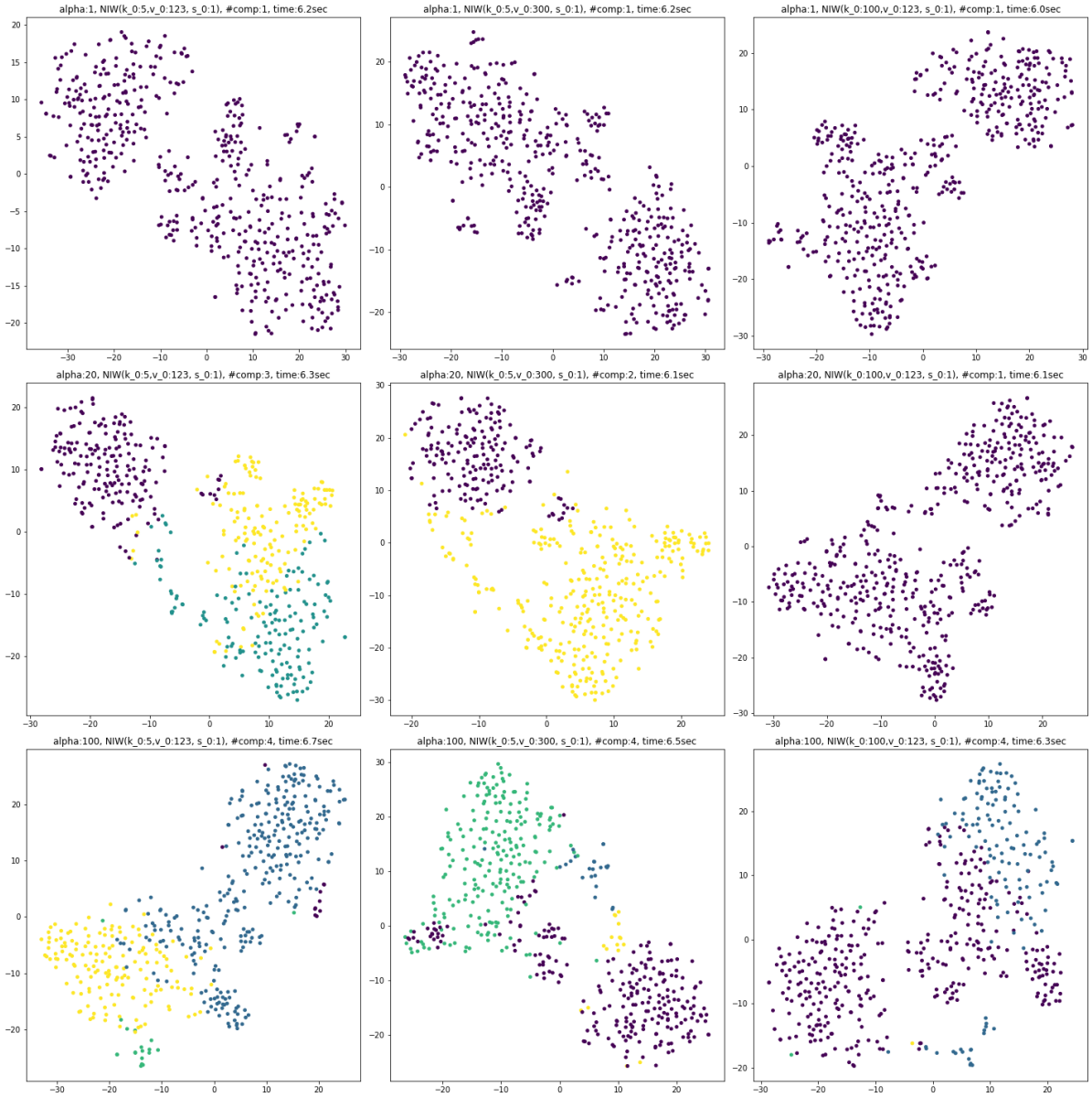

Figure 4: Experiments using different values for alpha and the priors. We can see the in general, increasing alpha(concentration parameter) have the effect of more clusters. However, it is affected also by the values of priors (see figure alpha: 20 and k0: 100)

Figure 5: Clustering results using Dirichlet Process Gaussian Mixture Models

Figure 6: Results with 5 iterations

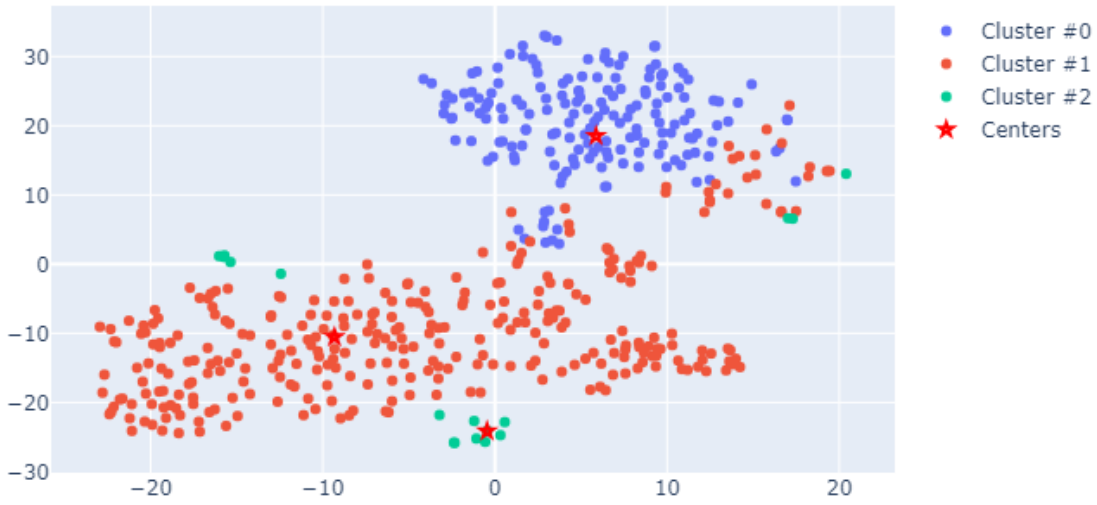

Figure 7: results with 20 iterations

5. **Does it scale ?**

   Our model is quite constraint in terms of dimensionalities and number of instances it can handle. we evaluated the time elapsed while running inference with respect to a range of dimensionalities and number of instances.

   (a) **Dimensionality based constraint**

   With $T = 100$ (number of iterations) and the number of instances set to $N = 1000$, we gradually increased the data dimensionality and observed the time elapsed as shown in figure below

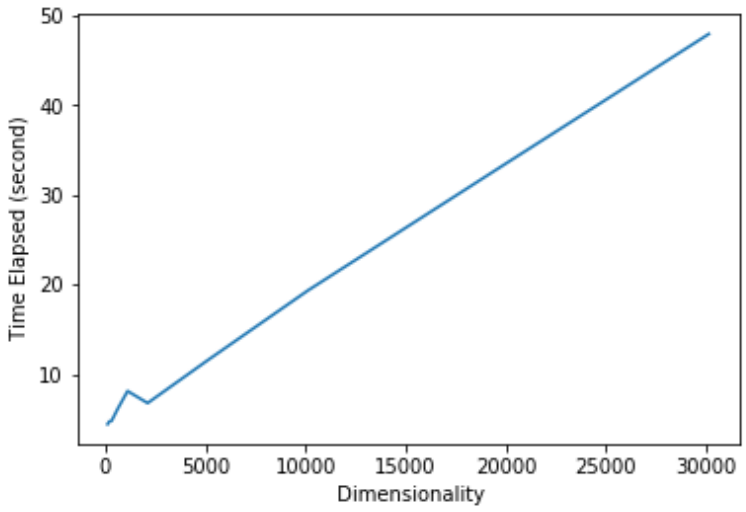

Figure 8: Time Elapsed with respect to an increasing number of features

Per the above figure, we remark the dimesionality increased linearly with the time elapsed . Using the Google Colab Environment with 12 GB RAM, we could not handle more than 30000 data points/features. Note that this upper bound could reduce as T and N are increased.

(b) **Data Volume constraint**

With $T = 100$ (number of iterations) and the number of instances set to $N = 1000$, we gradually increased the number of instances and observed the time elapsed as shown in figure below

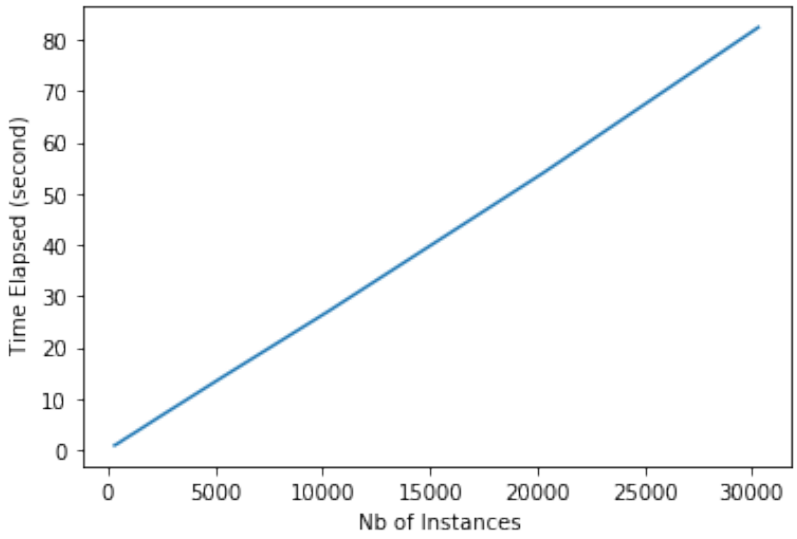

Figure 9: Time Elapsed with respect to an increasing number of instances

Per the above figure, we notice that the number of instances increase linearly with the time elapsed . Using the Google Colab Environment with 12 GB RAM, we could not handle more than 40000 instances . Note that this upper bound could reduce as T and N are increased.

6. **What would you try if you had to redo the project?** If we were to redo the project, there are some points we would like to try. They could qualify as well as future work and they can be classified in three main axis:

   (a) **A different representation of the points.** We know as a fact that the time-series would not qualify well as data to work on mainly because of their high dimensionality. Thus,at the moment we are using a Discrete Wavelet Transformation to extract meaningful features out of the raw signals and apply our model in the reduced 120 dimensional space. However, we believe that a Fourier Transformation of the signal could offer us another perspective over the data points and would be worth-trying.

   (b) **A different base distribution.** The great advantage the Dirichlet Process Mixture Model offers is the ability to use different base distributions, G_0. For the moment we are using a Normal distribution, which assumes the points can be clustered around centroids in a gaussian shape. However, we would like to try other base distributions to see if they offer a better approach over the obtained clusters.

   (c) **A different inference method.** For the moment we are using Collapsed Gibbs sampling and integrate out most of the parameters (π, μ, Σ). This proves to give an intuitive and easy implementation in terms of programming. However, the main drawback is that the priors must be conjugate priors of the distributions of the parameters. We may try as well other inference methods, namely Variational Inference, to check if the results differ and most importantly to check if we could overcome the running time-related scaling issues.

7. **References**

   For implementing this probabilistic model, we used as reference the following materials:

   (a) (Book) Machine learning: a probabilistic perspective, KP Murphy - 2012

   (b) (Video) Nonparametric Bayesian Methods: Models, Algorithms, and Applications I, II, III, https://www.youtube.com/watch?v=I7bgrZjoRhM

   (c) (Paper) Variational inference for Dirichlet process mixtures DM Blei, MI Jordan - 2006

   (d) (Code) For inspirational and debug purposes https://github.com/kamperh/bayes_gmm

   (e) (Tutorial) Dirichlet Process Mixture Models in Pyro, Uber Technologies, http://pyro.ai/examples/dirichlet_process_mixture.html - 2017