Dirichlet Process Mixture Models
Deliverable 1 - Problem Modeling

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

January 28, 2020

1. **Problem Intuition**

   The training dataset which was received from the competition organizers consists of 1677 training instances, which are in fact time-series of dimension 61440. They come from different flights and different sensors attached in many areas of Airbus Helicopters. The task is to perform anomaly detection on these time series, knowing that all the series in the training set are normal, while in the validation and the test set different anomalies might occur.

   The labels have been assigned in an unusual manner: if a part of a flight is anomalous, then the whole flight is flagged as anomalous. This implies that there can be time-series in the validation or the test set which, even though seem normal, might be flagged as anomalies.

   Starting from this observation, we believe that an advanced partitioning of the time series coming from different flights or from different periods of a flight might help in detecting these "normal" looking but anomaly-flagged points.

   Therefore, we aim to detect an underlying structure of the training examples coming from different flights. To achieve this, we propose a solution based on Gaussian Mixture Models, in which we don't set a priori the number of clusters sought. This is also known as a **Dirichlet Process Mixture Modelling** or **Infinite Gaussian Mixture Models**. This probabilistic model can be regarded as a normal mixture model with two key aspects:

   (a) K (i.e. the number of components) can be theoretically infinite.

   (b) The distribution of the models which compose the mixture is not bounded to being a Gaussian, it can be any distribution.

   We decided to use this kind of model because it will allow us to detect any number of clusters that compose the dataset and give the freedom of using any distribution, not necessarily a gaussian.

2. **Data preparation**

   Raw data are sequences of 61440 points. We need to reduce the dimension of the input. We are also interested in the frequency content of the sequences. The Fourier transform decomposes a signal in a sum of sine and cosine function. It allows to pass from time domain to frequency domain. The main inconvenient is that the information about when a frequency happens or disappears in time is lost. On the contrary, Wavelet Transform allows an analysis both in time and frequency, it can measure the evolution of frequency transients. The WT decomposes a signal into several scales representing different frequency bands. The Discrete Wavelet Transform produces a series of value matching the different scaling. From those series are extracted 120 features (including statistical values, entropy and zeros crossing statistics). Before proceeding to DWT, we are computing a rolling average on the sequence to reduce the number of points. Figure 1 shows the different steps of the data preprocessing.

Figure 1: Process for data preparation.

3. **Bayesian Network Representation**

In Figure 2, we present the graphical model for the Dirichlet Process Mixture Model. We assume that the base distribution is in fact a Gaussian, but the model can be easily enhanced with any other type of exponential distribution. We consider a prior α for the components vector π, and the G_0 is defined by the tuple of priors $\{\mu_0, \lambda_0, \Lambda_0, \nu_0\}$, which are specific priors for the Multivariate Normal Distribution.

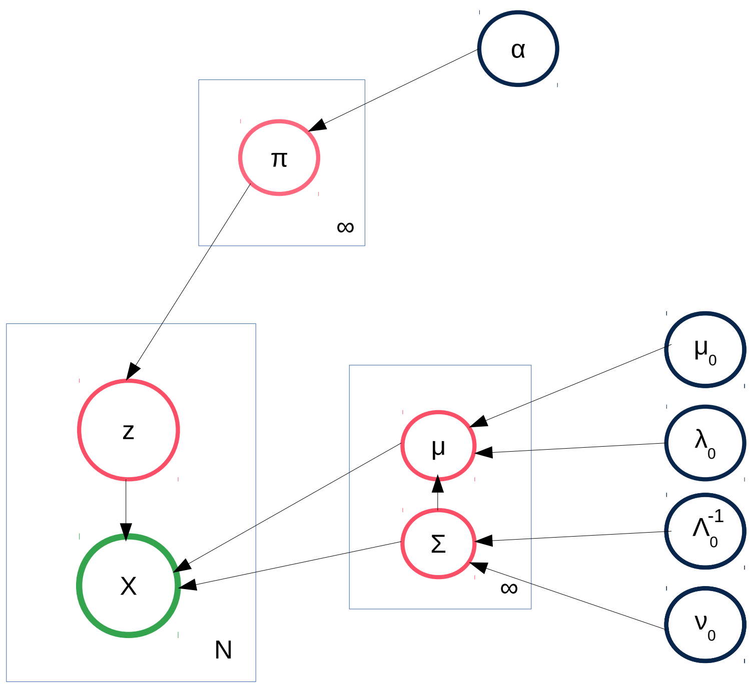

Figure 2: Bayesian Network Representation for the Dirichlet Process Mixture Model with Gaussian as the base distribution.

The important difference from the usual Gaussian Mixture Model is represented by the **possibly** infinite number of individual clusters defining the mixture. This will require special treatment during the inferation derivation and generation process, described in the following section.

4. **Generative Process**

Dealing with infinite dimensional probability vectors, the DPMM has a specific generative process.

(a) Generating $\pi = (\pi_1, \pi_2, ...\pi_k...)$ using a (**G**riffiths, **E**ngen, and **M**cCloskey) distribution: $\pi \sim GEM(\alpha)$, meaning:

$V_1 \sim Beta(1, \alpha) => \pi_1 = V_1$
$V_2 \sim Beta(1, \alpha) => \pi_2 = (1 - V_1)V_2$
...
$V_k \sim Beta(1, \alpha) => \pi_k = V_k \prod_{j=1}^{k-1}(1 - V_j)$
...

This trick is also known as the Stick Breaking Method, and can be observed in Figure 3. Basically it allows us to generate an infinite dimensional probability distribution vector, π whose values sum up to 1. In other words, it simulates an ∞-Dirichlet.

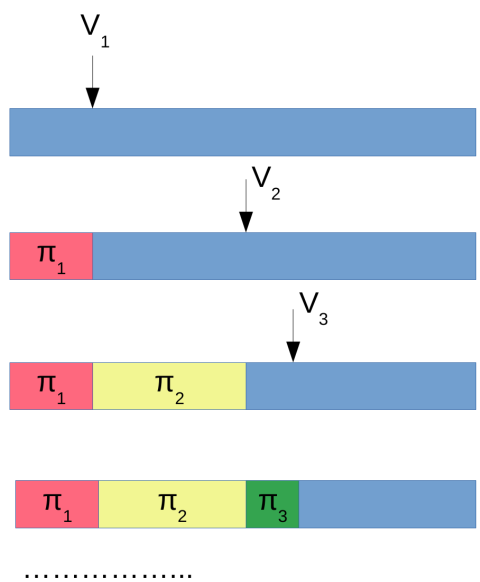

Figure 3: Stick Breaking Method. We generate a value for V, using the Beta distribution then compute the corresponding π value.

The generation of a value for π_k, happens **on demand**. This means that we keep generating values $V \sim Beta(1, \alpha)$ until we reached the k-th element.

(b) Generating $\Sigma = (\Sigma_1, \Sigma_2, ...\Sigma_k...), \forall k = 1 : \infty$

$\Sigma_k \sim InverseWishart(\Lambda_0^{-1}, \nu_0)$

(c) Generating $\mu = (\mu_1, \mu_2, ...\mu_k...), \forall k = 1 : \infty$

$\mu_k \sim \mathcal{N}(\mu_0, \frac{\Sigma_k}{\lambda_0})$

(d) Generating $z = (z_1, z_2, ...z_N), \forall n = 1 : N$

$z_n \sim Categorical(\pi)$

(e) Generating an element X_n

$X_n \sim \mathcal{N}(\mu_{z_n}, \Sigma_{z_n})$