A Bayesian Finite Mixture Model for Network-Telecommunication Data

Vasileios Manikas
A Bayesian Finite Mixture Model for Network-Telecommunication Data

Vasileios Manikas*

Abstract

A data modeling procedure called Mixture model, is introduced beneficial to the characteristics of our data. Mixture models have been proved flexible and easy to use, a situation which can be confirmed from the majority of papers and books which have been published the last twenty years. The models are estimated using a Bayesian inference through an efficient Markov Chain Monte Carlo (MCMC) algorithm, known as Gibbs Sampling. The focus of the paper is on models for network-telecommunication lab data (not time dependent data) and on the valid predictions we can accomplish. We categorize our variables (based on their distribution) in three cases, a mixture of Normal distributions with known allocation, a mixture of Negative Binomial Distributions with known allocations and a mixture of Normal distributions with unknown allocation.

Keywords: Mixture Model, Bayesian Inference, Markov Chain Monte Carlo, Gibbs Sampling, Network-Telecommunication Lab Data.

* E-mail: vasileios.manikas1@gmail.com
Acknowledgements

After two years of pure statistics, thousands of hours of reading books and papers my life as a master’s student is coming to an end. This paper is a direct outcome of all my professors’ efforts, these two years, to try to show us and put in our heads that statistics is not only graphs and numbers. First of all, I have to thank Matias Quiroz, the person who rocked my statistical world with the Bayesian way of life (it is your fault!!!). I would also like to thank my supervisor in Stockholm University Frank Miller for his patience and always great suggestions; my parents, the best parents of the world, I owe everything to them. Last but not least, my special thanks goes to Paul Stewart, my supervisor in Ericsson, the most supportive person I have ever met; he is the guy with whom we had lot of conversations about statistics the past four months and totally believes I am a lunatic (I hope we’ll continue our chatting in the future, cheers mate!!!).
Contents

1. Introduction .................................................................................................................. 5

2. Data ............................................................................................................................... 7
   2.1 Data collection procedure ....................................................................................... 7
   2.2 Variables .................................................................................................................. 7

3. Data preparation ........................................................................................................... 9
   3.1 Distribution investigations ..................................................................................... 9
   3.2 Outliers .................................................................................................................. 12
   3.3 Distribution fitting ................................................................................................. 13

4. Theoretical background-methodology ....................................................................... 16
   4.1 Theoretical background ......................................................................................... 16
      4.1.1 Finite Mixture Model .................................................................................... 16
      4.1.2 Bayes Theorem ........................................................................................... 17
      4.1.3 Gibbs Sampler ............................................................................................... 18
   4.2 Methodology ........................................................................................................... 19
      4.2.1 Mixture of Normal distributions (known allocation) ..................................... 20
      4.2.2 Mixture of Negative Binomial Distributions (known allocation) ............... 23
      4.2.3 Mixture of Normal distributions (unknown allocation) ............................... 24

5. Application ................................................................................................................... 28
   5.1 Application to Mixture of Normal Distributions (known allocation) ................. 28
   5.2 Application to Mixture of Binomial Distributions (known allocation) ............... 33
   5.3 Application to Mixture of Normal Distributions (unknown allocation) ............. 36
   5.4 Mixture with different hyper-parameters ............................................................. 40
   5.5 Visualization results .............................................................................................. 44

6. Conclusions .................................................................................................................... 46

References ......................................................................................................................... 47
Appendix ............................................................................................................................. 49
1. Introduction.

The topic of our research was motivated and supported by the Observability support team within the Product Integration and Release department at Ericsson AB. As a result, it is of great importance to provide some details about the objectives of this department and at the same time to understand why it intrigued our statistical interest.

The focus of this department is to assure the validity and functionality of new LTE 4G Radio software releases, prior to global release to telecom operators. One subset of the software testing procedure involves a 12-hour test case, where the system under test is subjected to heavy traffic load. During this test period various metrics related to the behavior and characteristics of the system are recorded (such as dropped calls, throughput, connected users etc.). The results from the new software tests are then compared with the results from a known baseline. A baseline corresponds to a similar test case that has previously been performed on the latest globally released customer software, hence is considered of high quality. Thus, comparing the new software test results to the baseline tests results, quality estimations of the new software can be used directly as part of the decision criteria as to whether the new software is ready to be released to external customers. In addition, a subset of the comparisons between the new software and the baseline software are achieved through hypothesis testing (t-test and Wilcoxon test).

One of the major issues encountered during the software testing procedure involves the complexity of the test environment. The test environment can be affected by many external factors (such as the large number of network elements and supporting network equipment, geographical distribution of network elements, radio environment conditions (interference) etc.). Furthermore, existing software validation methods only compare the new software results against one baseline period, which affects directly the percentage of Type I / Type II error from the hypothesis testing due to fluctuations in the test environment.

From our perspective, our major responsibility is to create a model which will consider the characteristics of previously validated baselines in a way that can account for the fluctuations of the complex test environment that exist in validated tests. As a result, our viewpoint is to create a “Superior Baseline” based on the results from multiple previous baselines. A finite mixture model is going to be introduced beneficial to the creation of the Superior Baseline.

A Bayesian approach is going to be followed for the creation of the mixture model through the use of Gibbs Sampling. We will assume prior distributions for our unknown parameters, with respect to the conjugacy principal through the likelihoods of our datasets. The reason why we are going to use this method is the fact that it is ambitious to take samples from our original distribution and by combining the Bayes theorem along with the Gibbs sampler we will achieve to generate samples from our posterior distributions.
However, this is not our ultimate goal. We want to change the current status because the procedures which have been already used do not seem proper. The t-test needs lot of assumptions to be fulfilled in order to be valid, which in the case of our dataset it is extremely difficult to occur. As soon as we obtain the generated samples from our posterior distributions, we will use them in order to compute the credible intervals for the mixture model. These credible intervals will include all the information from the previous baselines along with the fluctuations of the complex test environment. The outcome of this, is to use these credible intervals from the “Superior Baselines” and in the future to compare them with the intervals of the future software test-runs. If the intervals of the new software tests are included in the Superior Baseline credible intervals, then the test-run will be assumed successful, otherwise it will be faulty (or rather in need of further investigation).

All the results of this paper were obtained by using the software R. In addition, in section 4, packages “outliers” and “fitdistrplus” were used, in order to clean our datasets from the presence of outliers and to obtain the fitting. In section 5, package “gtools” was used for the Dirichlet distribution and “coda” for transforming the augmented samples from the Gibbs Sampler to MCMC† and also for the convergence and autocorrelation plots. Furthermore, all the plots in section 6, were constructed through “ggplot2”.

The structure of this thesis is as follows; section 2 includes all the statistical theory which was used for the research. In section 3, details about the data and the variables, which were used, are provided. Section 4, considers all the distributional assumptions we made for our datasets. In chapter 5, we include all the application steps which were followed, for the mixture models and computation of the credible intervals of the “Superior Baselines”. Finally, in section 6, we provide our conclusions from our research.

---

† Markov Chain Monte Carlo
2. Data

The four latest baselines for each variable will be used in this research, which include the valid measurements of four different baseline runs that occurred between 1st of December 2015 and 31st of January 2016.

Each baseline run contains 1068 different variables (counters), where each variable (counter) measures different objects during the data collection procedure. Taking under consideration the short period for producing this research we chose nine variables from the complete dataset. The choice of these nine variables was not random. After an extensive discussion with our supervisor from the Observability section of Product Integration and Release department of Ericsson AB, nine “important” variables (counters) where chosen. All variables are of course important since they measure different facts, however the set chosen were derived from vital key performance indicators (KPIs) formulas, which are always the first KPIs to be studied after a software test has been executed.

In the following sessions, 3.1 and 3.2, we provide a description of the data collection procedure and some definitions from the variables we are using.

2.1. Data collection procedure

The data for our research is obtained under a specific process which is described as follows; a software test case is executed for 12 hours, during which, different counters measure different characteristics and behaviors of the new software. Such characteristics can be the number of dropped calls, the maximum number of devices connected, the total number of mobile handovers between radio cells, etc. In each 12-hour test case, data is obtained every 15 minutes (with the recorded values reset every 15 minutes); hence 48 observations are made which are regarded to be independent. As a result, we are dealing with numerical count data.

The exact same procedure is used for a baseline test case, with the only difference being in the software loaded on the radio network equipment (already released software is used). After this procedure had occurred, we obtained four datasets (one dataset for each baseline) of 48 observations for each one of our variables.

2.2. Variables

The variables, which were used for the research, represent counters which count different facts during a 12-hour period run. A detailed description of each variable (counter) is given bellow:
**RrcConnEstabAtt**, the total number of RRC (radio resource control) connection request attempts. This counter measures the number of connection requests that have been received at a radio base station cell from UE equipment (i.e. Mobile Handset). It is a request to be granted access to the radio network.

**ErabRelAbnormalEnbAct**, the total number of abnormal E-RAB (radio access protocol) releases per cell initiated by the ENB where there was data in either the UL or DL buffer (i.e. active session). This counter is a measure of the number of LTE (4G) calls that have been disconnected abnormally by the radio base station cell while there has been data in the radio base station buffers, ready to be transmitted.

**SessionTimeUe**, accumulated active session (data transferred) time for all UEs in a cell. This is an accumulative counter that measures (per radio base station cell) the total accumulated active session time for all UEs (i.e. Mobile Handset). Active session time means data that has been transmitted in the uplink or downlink in the previous 100 milliseconds.

**ErabRelMMe**, the total number of E-RAB (radio access protocol) releases per cell initiated by the MME excluding successful handover. This counter is stepped when a call is released by the radio base station due to an order from the core network.

**CellHoPrepSuccLteIntraF**, the number of successful intra LTE intra frequency handover preparations (same frequency in source and destination cell). This counter measures the number of successful LTE Intra frequency handover preparations (The preparation phase is successful if the target cell for handover has indicated passes all the resource requirements for the handover request of the UE (i.e.: Mobile Handset)).

**PdcpPktReceivedDl**, the total number of DRB packets (PDCP SDUs) received by RBS in PDCP in the downlink. When carrier aggregation is used a PDCP SDU can be sent over multiple cells (PCell / SCell(s)). This counter measure the total number of data packets (PDCP, SDUs) received from the core network to be transmitted in the downlink direction to a UE device (i.e. Mobile Handset).

**PdcpPktLostUl**, the total number of DRB packets (PDCP SDUs) lost in the uplink. This counter measures the total data packets lost in the uplink direction at the base station. The number of packets lost is detected through sequence numbers.

**SchedActivityCellUl**, the aggregated number of ms (milliseconds) in which DRB data was required to be scheduled in the uplink. This counters measures the total time (milliseconds) in which data was required to be scheduled in the radio base station in the uplink direction.

**RrcConnMax**, the peak number of UEs in RRC connected mode. This is a gauge (watermark) counter, indicating the maximum number of connected devices in the radio base station cell during the sampling period.
3. Data preparation

One of our first assumptions, when we were introduced to the topic of this research, was that a Poisson distribution would fit proper in our datasets, depended on the fact that we would deal with numerical count data. Really soon, we realized that this was not the case. The majority of our variables included very big numbers (up to millions) and also the datasets had huge variances. In addition, the presence of outliers made our datasets more difficult to approach. The focus of which distribution fits best for each of our variables is described in section 4.1, while a discussion about outliers is following in section 4.2 and fitting in 4.3.

3.1. Distribution investigation

To begin with, we had to investigate which distribution fits best on each variable; we started by observing visually our data and more specifically by creating histograms and Q-Q plots. The following graphs represent histograms and Q-Q plots for the 4 baselines of the variable RrcConnEstabAtt.

Figure 1. Histograms for variable RrcConnEstabAtt. Each one represents a different baseline. Note that the range of the x-axis values differs.
Based on the visualization inspection of the previous graphs it is really difficult to understand how our data is distributed. On the contrary, instead of checking each 48-observation dataset separately it would be of interest to investigate them as one dataset of 192 observations. Taking under consideration the structure of mixture models, we have one dataset which we categorize with some weights into K components. A clear picture of how one of our datasets looks is presented in the following graphs.

Figure 2. Q-Q plots for variable RrcConnEstabAtt. Each one represents a different baseline.
After merging the datasets, it becomes more straightforward to decide how our data is distributed by providing us with guidance on which distributions we should consider that can have a proper fit.
3.2. Outliers

One of the first interesting outcomes after the visualization inspection from histograms and Q-Q plots is the presence of outliers in our dataset. Outliers have always been a topic for debate in the statistical community. The definition of outlier itself varies in the statistical literature. Hawkins (1980, pp.1) defined an outlier as “an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism”. Grubbs (1969) stated, “An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs”. The substance behind all different definitions is that an outlier is a data point which is located far from the ordinary value range of our dataset.

The presence of outliers in a dataset can have a negative impact in the statistical analysis for the following reasons: (1) they can increase the error variance and decrease power of test, (2) they can decrease normality, (3) they can seriously influence estimates that may be of substantive interest (Osborne & Overbay, 2004). There are two major origins responsible for the existence of outliers in a dataset, errors in data (human errors, such as measurement error, sampling error, etc.) and, the present variability in the dataset (Anscombe, 1960).

Outlier detection has to be of considerable emphasis in any statistical research. There are several different methods for detecting an outlier, depending on whether it is a univariate or multivariate case. For univariate outliers, the easiest and most of the times effective procedure is by visual inspection of the data. Another commonly used technique is based on the distance between data points and the mean, where a commonly used rule of thumb states that a data point with three or more standard deviations far from the mean is a possible outlier.

The primary issue arrives after the detection of the outliers, in the sense of how we should treat them. Throughout the statistical literature there has been a lot of arguing on what to do in the presence of outliers, one thing stands for granted, we have to do something with them. One option is to remove them completely, another is to transform our data (take the logs) or to truncate the data. From our perspective, the treatment of outliers is completely subjective and should always be based on the experience and reasoning from the researcher’s scope.

Our next step was to understand the nature of our outliers and the reason of their occurrence. With the support and help from the people in the Observability team, we made some conclusions on what possibly caused the outliers. One reason is related to human errors, the person responsible for a software test starts the procedure earlier than expected, before the traffic level has reached the expected load. However, the most common cause of outliers in our data is affected by environment or software issues, a traffic simulator having a temporary problem, a software malfunction (bug) or even radio environment fluctuation due to an external interferer.
In our case, our spotlight is to understand the meaning and importance of the baseline from the Observability’s department scope. A baseline carries all the information from valid already released software. As a result, our baselines should not contain outliers, since they represent the validity of the software that has been previously released to external customers, along with the fact that at the same time this software does not produce any faults. To sum up, getting under consideration the nature of our outliers in combination with the context of the baselines we precede our research by removing them.

### 3.3. Distribution fitting

After we have removed the outliers, we start the fitting procedure for our variables with distributions of known form. According to the Q-Q plots and histograms we had investigated before, a normal distribution would have a good fit in six of the nine selected variables. Some representative graphs, for the variable RrcConnEstabAtt, of the fitting of the normal distribution are the following.

![Figure 5. Representation of the fitting of a Normal distribution in the variable RrcConnEstabAtt.](image-url)
In addition to the variables which we fit a normal distribution, we performed a Kolmogorov – Smirnoff test as well. The Kolmogorov – Smirnoff test, apart from a non-parametric test for equality of continuous, one-dimensional probability distributions, can be used as a goodness of fit test. In the case for testing normality, all data points are standardized and compared with the standard normal distribution. The result of the Kolmogorov – Smirnoff test does not reject our belief that our samples come from Normal distribution. The results are represented in the following table.

Table 1. P-values for the Kolmogorov-Smirnoff test.

<table>
<thead>
<tr>
<th>Variables</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RrcConnEstabAtt</td>
<td>0.3165</td>
</tr>
<tr>
<td>SessionTimeUe</td>
<td>0.7547</td>
</tr>
<tr>
<td>CellHoPrepSuccLeIntraF</td>
<td>0.4602</td>
</tr>
<tr>
<td>PdcpPktReceivEdDI</td>
<td>0.7417</td>
</tr>
<tr>
<td>PdcpPktLossUl</td>
<td>0.9069</td>
</tr>
<tr>
<td>SchedActivityCellUl</td>
<td>0.9834</td>
</tr>
</tbody>
</table>

For the distribution of the variable ErabRelAbnormalEnbAct the assumption of normality is not at all valid. Our investigation based on the type of the data that our variable includes, suggested that we should fit a Poisson or a Negative Binomial distribution as well. Visually both distributions seem to fit in our data well; however, taking under consideration the AIC information criteria the Negative Binomial distribution was preferred. The graphs for the fitting are the following.

![Empirical and Theoretical Distributions](Diagram1.png)

![Empirical and Theoretical CDFs](Diagram2.png)

Figure 6. Representation of the fitting of Negative Binomial Distribution in the variable ErabRelAbnormalEnbAct.
Furthermore, in the case of the variables ErabRelMme and RrcConnMax things were not at all obvious. None of the known form distributions seemed to fit well enough for our dataset. According to the fact that we are interested in the inferences from the data along with the state that our datasets include huge values, we will treat these variables as a mixture of normal distributions by assuming their allocations to be unknown. Our reasoning for treating this variable in such a way was motivated from the following histograms.

Figure 7. Histogram of the variable ErabRelMme.

Figure 8. Histogram of the variable RrcConnMax.
4. Theoretical background-methodology

4.1. Theoretical background

In sections 4.1.1 - 4.1.3 it is described all the theoretical background which was taken under consideration for the creation of our models.

4.1.1. Finite Mixture Model

A finite mixture model (McLachlan & Peel, 2000) is a probabilistic model of the form:

\[ f(y) = \sum_{k=1}^{K} \omega_k \times f(y; \theta_k) \]

where,

- \( y \) represents our dataset which is a vector of \( n \) elements \((y_1, ..., y_n)\),
- \( k \) is the number of components in the mixture model \((k = 1, ..., K)\),
- \( \omega_k \) is the mixing weight, for all \( k \)'s \( \omega_k \geq 0 \) and \( \sum_{k=1}^{K} \omega_k = 1 \),
- \( f(y; \theta_k) \) is the probability density function of the \( k \)-th component,
- \( \theta \) is a vector of \( K \) elements which includes all our unknown parameters.

The finite mixture model provides a natural representation of heterogeneity in a finite number of latent classes, while it concerns modeling a statistical distribution by a mixture (or weighted sum) of other distributions. In other words, it is a combination of two or more probability density functions. Thus, a mixture model is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs (Wikipedia, 03/05/2016).

In most of the cases in, a mixture model is also described as missing data model (see Marin et al. 2005). In this case, one should define an indicator variable \( s_i \), \( 1 \leq i \leq n \), for each individual, that indicates from which component each observation belongs to. However, this auxiliary variable is not observable. As a result, the model can be seen as a hierarchical model, where on top there are the unknown parameters \( \theta \) \((\theta = (\theta_1, ..., \theta_K) )\), then the missing data which depend on \( \theta \), and at the bottom the observed data \( y \), which distribution depends on \( s \) and \( \theta, y \sim f(y|\theta,s) \) (see Diebolt & Robert, 1994). By unknown parameters we refer to characteristics of known form distributions, for example if our data was normally distributed our unknown parameters would be the mean and variance. Furthermore, a mixture model can be seen as a non-parametric model, due to their feature to approximate nonstandard distributions, a characteristic which make them extremely useful in situations where a known form distribution cannot describe a dataset.
One of the first approaches to mixture modeling was performed by Pearson (1894), where he used a mixture of two univariate normal distributions with unknown means and unknown variances. After this, not so much was done in mixture modeling, since it was extremely difficult to perform the appropriate calculations for the posterior distributions. Then, in the early 70’s when statistical problems started to become more and more complicated (e.g. datasets were becoming huge), in combination with the revolutionary use of Bayesian inference at this time, the scenery had started to change.

Finite mixture models are an extremely flexible method of modeling (McLachlan & Peel, 2000), they provide an interesting alternative to non-parametric modelling, while they are less restrictive than the usual distributional assumptions (Diebolt & Robert, 1994). Such characteristics have made the mixture models widely used the past 25 years. Actually, the use of computers after the late 80’s changed the whole statistical world and led to great progress in mixture modeling too. Everit & Hand (1981), Titterington et al. (1985) , Bernando & Giron (1988) are some of the most pioneer references in mixture modeling where they analyze deeply the mixture distributions and applicability of this type of statistical modeling.

4.1.2. Bayes Theorem

One of the most challenging parts comes with the estimation of our unknown parameters. Our approach is based on hierarchy which makes the use of Bayesian inference ideal for our purpose.

The Bayes theorem states:

\[
p(\theta|y) = \frac{p(\theta) \times p(y|\theta)}{p(y)} \propto p(\theta) \times p(y|\theta)
\]

Where, \( \theta = (\theta_1, \ldots, \theta_K) \), the unknown parameters vector (different for each component k), \( y \) is the dataset \((y_1, \ldots, y_n)\). \( p(\theta|y) \) stands for the posterior density, \( p(y|\theta) \) is our likelihood and \( p(\theta) \) is our prior distribution for the unknown parameters. The proportionality symbol states that, as \( \theta \) varies but keeping \( y \) fixed the left hand side is equal to a constant times the right hand site.

The reasoning behind Bayes theorem reveals at the same way the philosophy of Bayesian inference and how different is from the classical statistical approach. In other words, our data affects the posterior inference only through the likelihood while we express our state of knowledge about anything unknown with a probability distribution through our prior.
4.1.3. Gibbs Sampler

The Gibbs sampler is a technique for generating random variables from a (marginal) distribution indirectly, without having to calculate the density (see Casella & George 1992). The mechanism of Gibbs sampler is based on simulations. During this procedure what Gibbs sampler really does, is to generate a Markov chain of random variables, which finally converge to the distribution of interest $f(y)$.

But what is a Markov chain? (The following representation is influenced by the laboratory-lectures in Bayesian Inference course, by Matias Quiroz, autumn semester 2015) For simplicity consider a discrete sample space for $\theta$. Example:

$$\pi(\theta) = \begin{cases} a_1, & \text{if } \theta = \phi_1 \\ a_2, & \text{if } \theta = \phi_2, \ a_1 + a_2 + a_3 = 1 \\ a_3, & \text{if } \theta = \phi_3 \end{cases}$$

A Markov process is a collection of r.v’s $\{\theta^{(t)}\}_{t \geq 0}$ with the property

$$\Pr(\theta^{(t)} = \phi^{(t)} | \theta^{(t-1)} = \phi^{(t-1)}, \ldots, \theta^{(1)} = \phi^{(1)}) = \Pr(\theta^{(t)} = \phi^{(t)} | \theta^{(t-1)} = \phi^{(t-1)})$$

where, $\phi^{(t)}$ denotes the state of the process at period $t$.

In the example with three states above: $\phi^{(t)} \in \{\phi_1, \phi_2, \phi_3\} \forall t \geq 0$.

A sequence generated by a Markov process is often called a Markov chain.

Based on the previous property, a Markov chain is a random process which undergoes transitions from one state to another on a state space, where the probability distribution of the next state depends only on the current state and not on the sequence of events that preceded it. This characteristic of a Markov chain is called “memorylessness”.

Why to use the Gibbs sampler? In our model, the parameter vector is divided into $K$ components, $\theta = (\theta_1, \ldots, \theta_K), 1 \leq k \leq K$ and it is difficult to simulate from $\pi(\theta) = \pi(\theta_1, \ldots, \theta_K)$, although it is easy to simulate from the full conditional posteriors

$$\pi(\theta_1 | \theta_2, \theta_3, \ldots, \theta_K)$$
$$\pi(\theta_2 | \theta_1, \theta_3, \ldots, \theta_K)$$
$$\vdots$$
$$\vdots$$
$$\pi(\theta_K | \theta_1, \theta_2, \ldots, \theta_{K-1})$$

where, the Gibbs sampler simulates from $\pi(\theta)$ by alternating the full conditionals.
A representation of how Gibbs sampler operates is given in the following table:

**GIBBS SAMPLER**

Obtain N samples from $\pi(\theta)$.

- Set an (arbitrary) start point, 
  $$\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_K^{(0)})$$

- For $i = 1, \ldots, N$, repeat
  1. $\theta_1^{(i)}$ is randomly drawn from $\pi(\theta_1 | \theta_2^{(i-1)}, \theta_3^{(i-1)}, \ldots, \theta_K^{(i-1)})$,
  2. $\theta_2^{(i)}$ is randomly drawn from $\pi(\theta_2 | \theta_1^{(i)}, \theta_3^{(i-1)}, \ldots, \theta_K^{(i-1)})$,
  
  K. $\theta_K^{(i)}$ is randomly drawn from $\pi(\theta_K | \theta_1^{(i)}, \theta_2^{(i)}, \ldots, \theta_{K-1}^{(i)})$

*Note: In each draw, the latest update of each block is used.*

**Table 2. Representation of the operation of Gibbs Sampling** (Quiroz, 2015)

### 4.2. Methodology

Based on the distribution fitting from section 3.3, we define three different models that will be used in this paper, (i) a mixture of Normal distributions with known allocation, (ii) a mixture of Negative Binomial distributions with known allocations and (iii) a mixture of Normal distributions with unknown allocations.

After we investigate how our data is distributed, we are going to choose a prior distribution for each of the unknown parameters based on conjugacy. The property that the posterior distribution follows the same parametric form as the prior distribution is called conjugacy (Gelman et al., 2013). The most important arguments about conjugacy is the fact that by taking a conjugate prior, based on our likelihood, we know beforehand which distributional form our posterior will have and at the same time it makes our computations easier. Since we do not have any prior information for our data, hierarchical priors will be used in the cases of (i) and (iii), while a non-informative conjugate prior will be used in case (ii). In case (ii) the selection of the non-informative prior is done in the sense that our data is informative enough to turn uninformative prior into an informative posterior. (For further information about the choice of prior distribution in finite mixture models see Frühwirth-Schnatter, 2006, sec.3.2, pp.58)
4.2.1. Mixture of Normal distributions (known allocation).

Our model has the form:

\[ y_i, s_i | \mu, \sigma^2, \omega \sim \sum_{k=1}^{K} \omega_k N(\mu_k, \sigma_k^2), \quad \sum_{k=1}^{K} \omega_k = 1, \quad i = 1, \ldots, n \quad (1) \]

\( K \) is the number of mixture components \((k = 1, \ldots, K)\), \( s_i \) denotes the allocation (from which component each observation comes from) \( \mu = (\mu_1, \ldots, \mu_K)' \), \( \sigma^2 = (\sigma_1^2, \ldots, \sigma_K^2)' \) and \( \omega = (\omega_1, \ldots, \omega_K)' \). Our mixture model can also be considered as the following hierarchical model:

\[ y_i, s_i | \mu_k, \sigma_k^2 \sim N(\mu_k, \sigma_k^2), \quad \Pr(s_i = k | \mu_k, \sigma_k^2, \mu_0, \tau_0^2, \omega_k) = \omega_k \]

\[ \mu_k \sim N(\mu_0, \tau_0^2) \]

\[ \sigma_k^2 \sim \text{Scaled} - \text{Inv} - \chi^2(v_0, s_0^2) \]

\[ \mu_0 \sim N(\tilde{\mu}, \tilde{\tau}^2) \]

\[ \tau_0^2 \sim \text{Scaled} - \text{Inv} - \chi^2(v_0^*, s_0^{2*}) \]

\[ \omega_k \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K) \]

The posterior (assuming apriori that \( \mu, \sigma^2, \omega, \mu_0, \tau_0^2 \) are independent) based on Bayes theorem is:

\[ p(\mu, \sigma^2, \mu_0, \tau_0^2, \omega | y, s) \propto p(y, s | \mu, \sigma^2, \mu_0, \tau_0^2, \omega) \times p(\mu, \sigma^2, \mu_0, \tau_0^2, \omega) \]

\[ = p(y, s | \mu, \sigma^2, \mu_0, \tau_0^2, \omega) p(\mu) p(\sigma^2) p(\mu_0) p(\tau_0^2) p(\omega) \]

\[ = \left( \prod_{\{i,s_i=k\}} N(y_i | \mu_k, \sigma_k^2) \right) \left( \sum_{k=1}^{K} N(\mu_k | \mu_0, \tau_0^2) \right) p(\sigma_k^2) p(\mu_0) p(\tau_0^2) p(\omega_k) \]

\( n_k \), is the number of observations in component \( k \)

\( \bar{y}_k \), is the mean for the observations in component \( k \)

Updating \( \mu_k \) for any \( k \):

\[ p(\mu_k | y_i, \sigma_k^2, \mu_0, \tau_0^2) \propto \left( \prod_{\{i,s_i=k\}} N(y_i | \mu_k, \sigma_k^2) \right) p(\mu_k) \]

\[ \text{Pr}(s_i = k | \mu_k, \sigma_k^2, \mu_0, \tau_0^2, \omega_k) = \omega_k \] stands for the Multinomial model (see Frühwirth-Schnatter,2006, sec.2.3.4,pp.29).

\( \mu_0 \) is the mean and \( \tau_0^2 \) is the variance for the prior Normal distribution of \( \mu_k \).

\( v_0 \) are the degrees of freedom and \( s_0^2 \) is the scale parameter for the prior Scaled-inv-\( \chi^2 \) distribution of \( \sigma_k^2 \).

\( \tilde{\mu} \) is the mean and \( \tilde{\tau}^2 \) is the variance for the prior Normal distribution of \( \mu_0 \).

\( v_0^* \) are the degrees of freedom and \( s_0^{2*} \) is the scale parameter for the prior Scaled-inv-\( \chi^2 \) distribution of \( \sigma_k^2 \).

\( \alpha_1, \ldots, \alpha_k \) are the prior parameters for the Dirichlet distribution.
Our likelihood is Normal, for conjugacy we choose a Normal prior for \( p(\mu_k) \), \( \mu_k \sim N(\mu_0, \tau_0^2) \) and the posterior is of the form:

\[
p(\mu_k | y, \sigma_k^2, \mu_0, \tau_0^2) = N(\mu_k | \mu_{0n}, \tau_{0n}^2)
\]

where,

\[
\mu_{0n} = \tau_{0n}^2 \left( \frac{n_k}{\sigma_k^2} \bar{y}_k + \frac{\mu_0}{\tau_0^2} \right)
\]

\[
\tau_{0n}^2 = \left( \frac{n_k}{\sigma_k^2} + \frac{1}{\tau_0^2} \right)^{-1}
\]

Updating \( \sigma_k^2 \) for any \( k \):

\[
p(\sigma_k^2 | y, \mu_k) \propto \left( \prod_{i,s_i=k} N(y_i | \mu_k, \sigma_k^2) \right) p(\sigma_k^2)
\]

Our likelihood is Normal, for conjugacy we choose a Scaled-Inv-\( \chi^2 \) prior for \( p(\sigma_k^2) \), \( \sigma_k^2 \sim \text{Scaled} - \text{Inv} - \chi^2(v_0, s_0^2) \) and the posterior is of the form:

\[
p(\sigma_k^2 | y, \mu_k) = \text{Scaled} - \text{Inv} - \chi^2(\sigma_k^2 | v_n, s_n^2)
\]

where,

\[
v_n = v_0 + n_k
\]

\[
s_n^2 = \frac{1}{v_n} \left( \sum_{k=1}^{K} \sum_{i=1}^{n_k} (y_{ki} - \mu_k)^2 + v_0 s_0^2 \right)
\]

Updating \( \mu_0 \):

\[
p(\mu_0 | \mu, \tau_0^2) = \left( \prod_{k=1}^{K} N(\mu_k | \mu_0, \tau_0^2) \right) p(\mu_0)
\]

Our likelihood is Normal, for conjugacy we choose a Normal prior for \( p(\mu_0) \), \( \mu_0 \sim N(\bar{\mu}, \bar{\tau}^2) \) and the posterior is of the form:

\[
p(\mu_0 | \mu, \tau_0^2) = N(\mu_0 | \mu^*, \tau^{2*})
\]
where,
\[ \mu^* = \tau^{2*} \left( K \frac{\tilde{\mu}}{\tau_0^2} + \frac{\tilde{\mu}}{\tau^2} \right) \]
\[ \tau^{2*} = \left( K \frac{1}{\tau_0^2} + \frac{1}{\tau^2} \right)^{-1} \]

Updating \( \tau^2_0 \):  
\[ p(\tau^2_0 | \mu, \mu_0) \propto \prod_{K} N(\mu_k | \mu_0, \tau^2_0) p(\tau^2_0) \]

Our likelihood is Normal, for conjugacy we choose a Scaled-Inv-\( \chi^2 \) prior for \( p(\tau^2_0) \), \( \tau^2_0 \sim \text{Scaled - Inv} - \chi^2(v_0^*, s_0^{2*}) \) and the posterior is of the form:
\[ p(\tau^2_0 | \mu, \mu_0) = \text{Scaled} - \text{Inv} - \chi^2(\tau^2_0 | v_n^*, s_n^{2*}) \]

where,
\[ v_n^* = v_0^* + K \]
\[ s_n^{2*} = \frac{1}{v_n^*} \left( \sum_{K} (\mu_k - \mu_0)^2 + v_0^* s_0^{2*} \right) \]

Updating \( \omega_k \) for any \( k \):
\[ p(\omega_k | y_i, s_i) \propto \prod_{i=1}^{n_k} p(s_i | \omega_k) p(\omega_k) = (\omega_1^{n_1} \omega_2^{n_2} \cdots \omega_k^{n_k}) p(\omega_k) \]

The likelihood is proportional to Multinomial distribution, for conjugacy we choose a Dirichlet prior for \( p(\omega_k) \), \( \omega_k \sim \text{Dirichlet}(\alpha_1, ..., \alpha_k) \) and the posterior is of the form:
\[ p(\omega_k | y_i, s_i) \sim \text{Dirichlet}(\alpha_1 + n_1, ..., \alpha_k + n_k) \]

In the case of the weights \( \omega_k \) we will assume apriori specific values for the hyper-parameters \((\alpha_1, ..., \alpha_k)\) which would give us weights of the form \( \omega_1 < \cdots < \omega_k \). Our perspective is to weight more the latest baselines and less the oldest. Moreover, the major reason why we include the weights in our model, instead of using fixed values directly, is because we want to incorporate the uncertainty through the prior distribution (Frühwirth-Schnatter, 2006, sec. 2.3.4, pp.35).
4.2.2. Mixture of Negative Binomial distributions (known allocation).

Our model has the form:

\[ y_i, s_i, r_k | \pi, \omega \sim \sum_{k=1}^{K} \omega_k \text{NBin}(r_k, \pi_k), \quad \sum_{k=1}^{K} \omega_k = 1, \quad i = 1, \ldots, n \quad (2) \]

\( K \), is the number of mixture components \((k = 1, \ldots, K)\), \( s_i \) denotes the allocation (from which component each observation comes from) \( \pi = (\pi_1, \ldots, \pi_K)' \), is the success probability for each component and \( \omega = (\omega_1, \ldots, \omega_K)' \), are the weights for the mixture. The parameter \( r_k \) \((r = (r_1, \ldots, r_K)')\), denotes the number of failures; an estimate after fitting a negative binomial distribution in each component was used for our model. Our mixture model can also be considered as the following hierarchical model:

\[ y_i, s_i, r_k | \pi_k \sim \text{NBin}(r_k, \pi_k) \]

*** \( \text{Pr}(s_i = k | \mu_k, \sigma^2_k, \mu_0, \tau^2_0, \omega_k) = \omega_k \)

\( \pi_k \sim \text{Beta}(\alpha, \beta) \)

\( \omega_k \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k) \)

The posterior (assuming apriori that \( \pi, \omega \) are independent) based on Bayes theorem is:

\[ p(\pi, \omega | y, s, r) \propto p(y, s, r | \pi, \omega) p(\pi, \omega) \]

\[ = p(y, s, r | \pi) p(\pi) p(\omega) \]

\[ = \left( \prod_{i, s_i = k} \text{NBin}(y_i | r_k, \pi_k) \right) p(\pi_k) p(\omega_k) \]

\( n_k \), is the number of observations in component \( k \)

Updating \( \pi_k \) for any \( k \):

\[ p(\pi_k | y_i, r_k) \propto \left( \prod_{i, s_i = k} \text{NBin}(y_i | r_k, \pi_k) \right) p(\pi_k) \]

*** \( \text{Pr}(s_i = k | \mu_k, \sigma^2_k, \mu_0, \tau^2_0, \omega_k) = \omega_k \) stands for the Multinomial model (see Frühwirth-Schnatter, 2006, sec.2.3.4, pp.29)
Our likelihood is Negative Binomial, for conjugacy we choose a Beta prior for \( p(\pi_k) \), \( \pi_k \sim Beta(\alpha, \beta) \) and the posterior is of the form:

\[
p(\pi_k | y_i, n_k) = Beta(\alpha^*, \beta^*)
\]

where,

\[
\alpha^* = \alpha + \sum_{i=1}^{n_k} y_{ki}
\]

\[
\beta^* = \beta + n_k \cdot n_k
\]

Updating \( \omega_k \) for any k:

\[
p(\omega_k | y_i, s_i) \propto \left( \prod_{i=1}^{n_k} p(s_i | \omega_k) \right) p(\omega_k) = (\omega_1^{n_1} \omega_2^{n_2} \ldots \omega_K^{n_K})p(\omega_k)
\]

The likelihood is proportional to Multinomial distribution, for conjugacy we choose a Dirichlet prior for \( p(\omega_k) \), \( \omega_k \sim Dirichlet(\alpha_1, \ldots, \alpha_K) \) and the posterior is of the form:

\[
p(\omega_k | y_i, s_i) \sim Dirichlet(\alpha_1 + n_1, \ldots, \alpha_K + n_K)
\]

(In the case of the weights \( \omega_k \) we behave the same way as in section 4.2.1.)

### 4.2.3. Mixture of Normal Distributions (unknown allocation)

Our model has the form:

\[
y_i | \mu, \sigma^2, k \sim \sum_{k=1}^{K} \omega_k N(\mu_k, \sigma_k^2), \quad \sum_{k=1}^{K} \omega_k = 1, \quad i = 1, \ldots, n
\]

Another formulation of the above model, by taking under consideration the unknown allocations is:

\[
y_i | \mu_k, \sigma_k^2, s_i = k \sim N(\mu_k, \sigma_k^2)
\]

\[
Pr(s_i = k | \omega_k) = \omega_k
\]

\( K \), is the number of mixture components (\( k = 1, \ldots, K \)), \( s_i \) denotes the allocation (from which component each observation comes from) \( \mu = (\mu_1, \ldots, \mu_k)' \), \( \sigma^2 = (\sigma_1^2, \ldots, \sigma_K^2)' \) and \( \omega = (\omega_1, \ldots, \omega_K)' \). Our mixture model can also be considered as the following hierarchical model:
\[ y_i | \mu_k, \sigma_k^2, s_i = k \sim N(\mu_k, \sigma_k^2), \quad \Pr(s_i = k | \omega_k) = \omega_k \]
\[ \mu_k \sim N(\mu_0, \tau_0^2) \]
\[ \sigma_k^2 \sim \text{Scaled} - \text{Inv} - \chi^2(\nu_0, s_0^2) \]
\[ \mu_0 \sim N(\bar{\mu}, \bar{\tau}^2) \]
\[ \tau_0^2 \sim \text{Scaled} - \text{Inv} - \chi^2(\nu_0^*, s_0^{2*}) \]
\[ s_i \sim \text{Multinomial} \]
\[ \omega_k \sim \text{Dirichlet}(a_1, ..., a_k) \]

The posterior (assuming apriori that \( \mu, \sigma^2, \omega \) are independent) based on Bayes theorem is:

\[
p(\mu, \sigma^2, s, \omega | y) \propto p(y | \mu, \sigma^2, s, \omega) p(\mu, \sigma^2, s, \omega) = p(y | \mu, \sigma^2, s) p(\sigma^2 | s, \omega) p(\omega) \]

\[
= \left( \prod_{i=1}^{n} p(y_i | \mu_{s_i}, \sigma_{s_i}^2, s_i) \right) p(\mu_k) p(\sigma_k^2) \left( \prod_{i=1}^{n} p(s_i | \omega_k) \right) p(\omega)
\]

\( n_k \), is the number of observations in component \( k \)
\( \bar{y}_k \), is the mean for the observations in component \( k \)

Updating \( \mu_k \) for any \( k \):

\[
p(\mu_k | y_i, \sigma_k^2, \mu_0, \tau_0^2) \propto \left( \prod_{i,s_i=k} N(y_i | \mu_k, \sigma_k^2) \right) p(\mu_k)
\]

Our likelihood is Normal, for conjugacy we choose a Normal prior for \( p(\mu_k), \mu_k \sim N(\mu_0, \tau_0^2) \) and the posterior is of the form:

\[
p(\mu_k | y_i, \sigma_k^2, \mu_0, \tau_0^2) = N(\mu_k | \mu_{0n}, \tau_{0n}^2)
\]

where,

\[
\mu_{0n} = \tau_{0n}^2 \left( \frac{n_k}{\sigma_k^2} \bar{y}_k + \frac{\mu_0}{\tau_0^2} \right)
\]

\[
\tau_{0n}^2 = \left( \frac{n_k}{\sigma_k^2} + \frac{1}{\tau_0^2} \right)^{-1}
\]
Updating $\sigma_k^2$ for any $k$:

$$p(\sigma_k^2|y_i, \mu_k) \propto \left( \prod_{i,s_i=k} N(y_i|\mu_k, \sigma_k^2) \right) p(\sigma_k^2)$$

Our likelihood is Normal, for conjugacy we choose a Scaled-Inv-$\chi^2$ prior for $p(\sigma_k^2)$, $\sigma_k^2 \sim \text{Scaled - Inv - } \chi^2(v_0, s_0^2)$ and the posterior is of the form:

$$p(\sigma_k^2|y_i, \mu_k) = Scaled - \text{Inv - } \chi^2(\sigma_k^2|v_n, s_n^2)$$

where,

$$v_n = v_0 + n_k$$

$$s_n^2 = \frac{1}{v_n} \left( \sum_{k=1}^{K} \sum_{i=1}^{n_k} (y_{ki} - \mu_k)^2 + v_0 s_0^2 \right)$$

Updating $\mu_0$:

$$p(\mu_0|\mu, \tau_0^2) = \left( \prod_{k=1}^{K} N(\mu_k|\mu_0, \tau_0^2) \right) p(\mu_0)$$

Our likelihood is Normal, for conjugacy we choose a Normal prior for $p(\mu_0)$, $\mu_0 \sim \mathcal{N}(\tilde{\mu}, \tilde{\tau}^2)$ and the posterior is of the form:

$$p(\mu_0|\mu, \tau_0^2) = \mathcal{N}(\mu_0|\mu^*, \tau^{2*})$$

where,

$$\mu^* = \tau^{2*} \left( \frac{K}{\tau_0^2} \bar{\mu} + \frac{\tilde{\mu}}{\tilde{\tau}^2} \right)$$

$$\tau^{2*} = \left( \frac{K}{\tau_0^2} + \frac{1}{\tilde{\tau}^2} \right)^{-1}$$

Updating $\tau_0^2$:

$$p(\tau_0^2|\mu, \mu_0) \propto \left( \prod_{k=1}^{K} N(\mu_k|\mu_0, \tau_0^2) \right) p(\tau_0^2)$$
Our likelihood is Normal, for conjugacy we choose a Scaled-Inv-$\chi^2$ prior for $p(\tau_0^2)$, $\tau_0^2 \sim Scaled - Inv - \chi^2(v_0^*, s_0^*)$ and the posterior is of the form:

$$p(\tau_0^2 | \mu, \mu_0) = Scaled - Inv - \chi^2(\tau_0^2 | v_n^*, s_n^*)$$

where,

$$v_n^* = v_0 + K$$

$$s_n^{2*} = \frac{1}{v_n^*} \left( \sum_{k=1}^{K} (\mu_k - \mu_0)^2 + v_0^* s_0^{2*} \right)$$

Updating $s_i$ for any $i$:

$$p(s_i = k | y_i, \mu_k, \sigma_k^2, \omega_k) \propto p(y_i | \mu_k, \sigma_k^2, s_i = k) p(s_i = k | \omega)$$

The above expression can be recognized as a Multinomial distribution with $K$ categories, as a result:

$$s_i | y_i, \mu_k, \sigma_k^2, \omega_k \sim Multinomial\left(1, \phi_1^{(i)}, ..., \phi_K^{(i)}\right)$$

where,

$$\phi_k^{(i)} = \frac{p(y_i | \mu_k, \sigma_k^2, s_i = k) \omega_k}{\sum_{k=1}^{K} p(y_i | \mu_k, \sigma_k^2, s_i = k) \omega_k}$$

Updating $\omega_k$ for any $k$:

$$p(\omega_k | y_i, s_i) \propto \left( \prod_{i=1}^{n_k} p(s_i | \omega_k) \right) p(\omega_k) = (\omega_1^{n_1}, \omega_2^{n_2}, ..., \omega_K^{n_K}) p(\omega_k)$$

The likelihood is proportional to Multinomial distribution, for conjugacy we choose a Dirichlet prior for $p(\omega_k)$, $\omega_k \sim Dirichlet(a_1, ..., a_k)$ and the posterior is of the form:

$$p(\omega_k | y_i, s_i) \sim Dirichlet(a_1 + n_1, ..., a_k + n_k)$$
5. Application

In total there are nine variables of which six (RrcConnEstabAtt, SessionTimeUe, CellHoPrepSuccLteIntraF, PdcpPktReceivedDl, PdcpPktLostUl, SchedActivityCellUl) are modeled using the mixture of Normal distributions with known allocation, one (ErabRelAbnormalEnbAct) is modelled by the mixture of Negative Binomial distributions and the rest two (ErabRelMme, RrcConnMax) by the mixture of Normal distributions with unknown allocation.

As it was described in section 3, the four latest baselines will be used in our models, as a result the number of components in our models will be four (K=4). Only in the case of the mixture of Negative binomial distributions for the variable ErabRelAbnormalEnbAct the number of components will be three (K=3), since the first baseline (oldest) was totally faulty and will not be considered in the mixture model.

5.1. Application of the mixture of Normal Distributions (known allocation).

Model (1) from section 4.2.1 was used for the application of mixture of Normal distributions with known allocations for the variables RrcConnEstabAtt, SessionTimeUe, CellHoPrepSuccLteIntraF, PdcpPktReceivedDl, PdcpPktLostUl, SchedActivityCellUl. The following values were used for the application of our model to the variables of this section.

**Number of components:**
K=4

**Hyper-parameters:**
\[ \tilde{\mu} = 0, \]
\[ \tilde{\tau}^2 = 1, \]
\[ v_0 = v_0^* = 5, \]
\[ s_0^2 = s_0^2^* = 10, \]
\[ \alpha_1 = 0, \alpha_2 = 20, \alpha_3 = 40, \alpha_4 = 60 \]

**Starting values:**
\[ \sigma_k^2 = (10,10,10,10)', \mu_0 = 0, \tau_0^2 = 1 \]

**Number of iterations for the Gibbs sampler:** 10,000

**Burn-in period:** 1,000

After using the model from section 4.2.1 and building the code in R for the Gibbs sampling, we obtained 10,000 samples for each one of the parameters in the model. Of great interest is to investigate whether our chains are converging and if there is any correlation between our sample points in the chains. Figures 9 and 10 provide the convergence and autocorrelation plots, after the burn-in period of 1,000 samples, for the variable RrcConnEstabAtt.
Figure 10 provides the autocorrelation plots for each one of our parameters, high autocorrelation within chains indicate slow mixing and slow convergence, a state which is not present in our chains. This can also be confirmed from figure 9, where after the burn in period our chains converge almost immediately.

Our next step includes the procedure where we get sample from the predictive distribution of the mixture model, from which we will compute the credible intervals. In order to obtain a sample from the predictive distribution of the mixture model, first we sample from the multinomial distribution with probabilities equal to the weights we had obtained from the Gibbs Sampler, to specify the allocations for each sample point. After, we sample from Normal distribution with mean and variance indicated by the allocation for each sample point. Figure 11 represents the histogram of the original dataset and the histogram of the predictive distribution (which includes 9,000 samples) of the mixture model after the augmented data procedure through the Gibbs sampler.

Finally, we compute the 99% credible intervals for the predictive distribution of the mixture model, which will be used to make comparisons to different software test cases, to test and check how well it operates. The 99% credible intervals for the variable RrcConnEstabAtt are (301677.2, 309170.8).
Figure 9. Convergence plots for the parameters $\mu_k$, $\sigma_k^2$, $\omega_k$, $\mu_0$, $\tau_0^2$ of the variable $RrcConnEstabAtt$. All the Markov chains converge proper. The black line represents the mean of each of our parameters and the dotted lines are the 2.5% and 97.5% quantiles of the convergence (9,000 samples for each parameter).
Figure 10. Autocorrelation plots for the parameters $\mu_k, \sigma^2_k, \omega_k, \mu_0, \tau^2_0$ of the variable RrcConnEstabAtt. Almost zero autocorrelation in our chains (9,000 samples for each parameter).
Figure 11. The left histogram represents our original dataset (190 data points) after we merged the four baselines. The right histogram represents the distribution of the mixture model after we augmented data through the Gibbs sampler (9,000 data points).

The convergence and autocorrelation plots for the rest of the variables (for this section) were almost the same (compared to convergence and autocorrelation plots for variable RrcConnEstabAtt) and are not presented in this paper. The histograms of the predictive distributions for the rest of the variables are provided in the Appendix. The computed credible intervals for the rest of the variables along with the variable RrcConnEstabAtt are presented in the table 3.

Table 3.

<table>
<thead>
<tr>
<th>Variable</th>
<th>99% Credible Intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.005%</td>
</tr>
<tr>
<td>RrcConnEstabAtt</td>
<td>301,677.2</td>
</tr>
<tr>
<td>SessionTimeUe</td>
<td>1,574,949</td>
</tr>
<tr>
<td>CellHoPrepSuccLteIntraF</td>
<td>11,851.31</td>
</tr>
<tr>
<td>PdcpPktReceivedDl</td>
<td>93,722,485</td>
</tr>
<tr>
<td>PdcpPktLostUl</td>
<td>57,370.83</td>
</tr>
<tr>
<td>SchedActivityCellUl</td>
<td>16,358,732</td>
</tr>
</tbody>
</table>
5.2. Mixture of Negative Binomial Distributions (known allocations).

Model (2) from section 4.2.2 was used for the application of mixture of Negative Binomial distributions with known allocations for the variable ErabRelAbnormalEnbAct. The following values were used for the application of our model.

Number of components:
K=4

Hyper-parameters:
\[ a = \left( \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right)' \]
\[ \beta = \left( \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right)' \]
\[ \alpha_1 = 0, \alpha_2 = 10, \alpha_3 = 25 \]

Starting values:
\[ r_1 = 2.41, \]
\[ r_2 = 2.83, \]
\[ r_3 = 8.57 \]

Number of iterations for the Gibbs sampler: 10,000
Burn-in period: 1,000

After using the above hierarchy, we built our code in R for the Gibbs sampler, from which we obtained 10,000 samples for each one of our parameters in the model. Of great interest is to investigate whether our chains are converging and if there is any correlation between our sample points in the chains. The following figures provide the convergence and autocorrelation plots, after the burn-in period of 1,000 samples, for the variable ErabRelAbnormalEnbAct.
Figure 11. Convergence plots for the parameters $\pi_k, \omega_k$ of the variable ErabRelAbnormalEnbAct. All the Markov chains converge proper. The black line represents the mean of each of our parameters and the dotted lines are the 2.5% and 97.5% quantiles of the convergence (9,000 samples for each parameter).

Figure 13. Autocorrelation plots for the parameters $\pi_k, \omega_k$ of the variable ErabRelAbnormalEnbAct. Almost zero autocorrelation in our chains (9,000 samples for each parameter).
Figure 12, provides the autocorrelation plots for each one of our parameters, high autocorrelation within chains indicate slow mixing and slow convergence, a state which is not present in our chains. This can also be confirmed from figure 11, where after the burn in period our chains converge almost immediately.

In order to obtain sample from the predictive distribution, we follow the same procedure as in 5.1. The following figures represent the histogram of the original dataset and the histogram of the predictive distribution (which includes 9,000 samples) of the mixture model after the augmented data procedure through the Gibbs sampler.

Finally, we can compute the 99% credible intervals for the predictive distribution of the mixture model, which will be used to make comparisons with the runs for the different types of software that want to test and check how well it operates. The 99% credible intervals for the variable ErabRelAbnormalEnbAct are presented in table 4.

Table 4.

<table>
<thead>
<tr>
<th>99% Credible Intervals</th>
<th>0.005%</th>
<th>0.995%</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErabRelAbnormalEnbAct</td>
<td>0</td>
<td>36</td>
</tr>
</tbody>
</table>
5.3. Mixture of Normal Distributions (unknown allocation).

Model (3) from section 4.2.3 was used for the application of mixture of Normal distributions with unknown allocations for the variables ErabRelMme and RrcConnMax. The following values were used for the application of our model to the variables of this section.

**Number of components:**
K=4

**Hyper-parameters:**
- $\tilde{\mu} = 0$,
- $\tilde{\tau}^2 = 1$,
- $v_0 = v_0^* = 5$,
- $s^2_0 = s^2_0^* = 10$,
- $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 10$

**Starting values:**
- $\sigma_k^2 = (10,10,10,10)'$,
- $\omega_k = (0.1,0.2,0.3,0.4)'$,
- $\mu_0 = 0$,
- $\tau_0^2 = 1$

**Number of iterations for the Gibbs sampler:** 12,000
**Burn-in period:** 2,000

Taking under consideration the above hierarchy, we built our code in R for the Gibbs sampler, from which we obtained 12,000 samples for each one of our parameters in the model. Of great interest is to investigate whether our chains are converging and if there is any correlation between our sample points in the chains. The following figures provide the convergence and autocorrelation plots, after the burn-in period of 2,000 samples, for the variable ErabRelMme.
Figure 15. Convergence plots for the parameters $\mu_k, \sigma_k^2, \omega_k, \mu_0, \tau_0^2$ of the variable ErabRelMme. All the Markov chains converge proper. The black line represents the mean of each of our parameters and the dotted lines are the 2.5% and 97.5% quantiles of the convergence (10,000 samples for each parameter).
Figure 16. Autocorrelation plots for the parameters $\mu_k$, $\sigma_k^2$, $\omega_k$, $\mu_0$, $\tau_0^2$ of the variable ErabRelMme. Small autocorrelation in our chains until lag=20 (10,000 samples for each parameter).

Figure 16, provides the autocorrelation plots for each one of our parameters, high autocorrelation within chains indicate slow mixing and slow convergence. It can be clearly observed that for parameters $\mu_k$, $\sigma_k^2$, $\omega_k$ exists a small amount of autocorrelation until lag=20. This can also be confirmed from figure 15, where after the burn in period our chains converge better after (maximum) 1000 iterations.
In order to obtain sample from the predictive distribution, we follow the same procedure as in 5.1. The following figure represents the histogram of the original dataset and the histogram of the predictive distribution (which includes 10,000 samples) of the mixture model after the augmented data procedure through the Gibbs sampler.

![Histogram of the dataset](image1.png) ![Histogram of the predictive distribution](image2.png)

*Figure 17. The left histogram represents our original dataset (190 data points) after we merged the four baselines. The right histogram represents the predictive distribution of the mixture model after we augmented data through the Gibbs sampler (10,000 data points).*

Finally, we can compute the 99% credible intervals for the distribution of the mixture model, which will be used to make comparisons with the runs for the different types of software that want to test and check how well it operates. The 99% credible intervals for the variable ErabRelMme are (10264.6, 14102.6).

The convergence and autocorrelation plots for the other variable (of this section) were almost the same (compared to convergence and autocorrelation plots for variable ErabRelMme) and are not presented in this paper. The histogram of the predictive distributions for the variable RrcConnMax is provided in the Appendix. The computed credible intervals for the variables ErabRelMme and RrcConnMax are presented in the table 5.

<table>
<thead>
<tr>
<th></th>
<th>99% Credible Intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.005%</td>
</tr>
<tr>
<td>ErabRelMme</td>
<td>10,264.6</td>
</tr>
<tr>
<td>RrcConnMax</td>
<td>6,535.829</td>
</tr>
</tbody>
</table>
5.4. A mixture with different hyper-parameters

Of great interest would be the case to investigate the behavior of our models in the case where we are setting new values for the hyper-parameters in the Gibbs sampler. The following results represent the same procedure as in 4.2.1, for the variable RrcConnEstabAtt, by setting our hyper-parameters as follows: $\tilde{\mu} = 100, \tilde{\tau}^2 = 1000, \nu_0 = 10, \nu^*_0 = 8, s^2_0 = 20, s^2_0^* = 50$. The reason why we do not change the hyper-parameters for the weights is the fact that we want to maintain the same weighting in the model along with the fact that this parameter is independent of the mean and variance (the starting values, number of K components, number of iterations and burn in period are the same as in section 5.1).

Figures 18 and 19 provide the convergence and autocorrelation plots after the burn-in period of 1,000 samples, for the variable RrcConnEstabAtt. From these figures we can observe that our chains converge proper and fast, since the autocorrelation is almost zero. Compared to figures 9 and 10, they seem to be almost identical with the only difference to be occurred in the convergence of the mean of the parameter $\mu_0$, where in figure 9 it converges to 0 and in figure 18 it converges to 100.

The following table presents the means of our parameters after the burn-in period (1,000 samples) for the variable RrcConnEstabAtt for the sections 5.1 and 5.4.

<table>
<thead>
<tr>
<th>Component K</th>
<th>$K=1$</th>
<th>$K=2$</th>
<th>$K=3$</th>
<th>$K=4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>section 5.1</td>
<td>$\mu_k$</td>
<td>305312.9</td>
<td>303753.6</td>
<td>306504.9</td>
</tr>
<tr>
<td>section 5.4</td>
<td></td>
<td>305314.2</td>
<td>303757.7</td>
<td>306503.1</td>
</tr>
<tr>
<td>section 5.1</td>
<td>$\sigma^2_k$</td>
<td>946248.2</td>
<td>974542.3</td>
<td>617161.6</td>
</tr>
<tr>
<td>section 5.4</td>
<td></td>
<td>891745.1</td>
<td>920150.6</td>
<td>582869.5</td>
</tr>
<tr>
<td>section 5.1</td>
<td>$\omega_k$</td>
<td>0.1516862</td>
<td>0.2191603</td>
<td>0.2839537</td>
</tr>
<tr>
<td>section 5.4</td>
<td></td>
<td>0.1514218</td>
<td>0.2198060</td>
<td>0.2836715</td>
</tr>
</tbody>
</table>

Based on the results from table 3, we can observe that the parameters $\mu_k$ and $\omega_k$ are almost identical in both cases in contrast to the parameter $\sigma^2_k$ where there exists an obvious difference. Even after we have changed the hyper-parameters for $\mu_k$ the mean for each of our components converges in the same space. As for the parameter $\sigma^2_k$, since we sample from different $Inv - \chi^2$ distributions (with larger hyper-parameters, compared to section 5.1) along with the fact that the variability in our dataset is huge leads the variance in each component to become smaller. This can also be confirmed from the histograms, in figure 20, for the predictive distributions of our mixture models.
Figure 18. Convergence plots for the parameters $\mu_k, \sigma_k^2, \omega_k, \mu_0, \tau_0^2$ of the variable $RrcConnEstabAtt$. All the Markov chains converge proper. The black line represents the mean of each of our parameters and the dotted lines are the 2.5% and 97.5% quantiles of the convergence (9,000 samples for each parameter).
Figure 19. Autocorrelation plots for the parameters $\mu_k, \sigma_k^2, \omega_k, H_0, \tau_k^2$ of the variable RrcConnEstabAtt. Almost zero autocorrelation in our chains (9,000 samples for each parameter).
Figure 20. The left histogram represents the predictive distribution of the mixture model for section 5.1 and the right histogram represents the predictive distribution of the mixture model with the hyper-parameters which were chosen in section 5.4.

Finally, the same status can be certified in the computation of the credible intervals which is presented in the following table.

Table 6.

<table>
<thead>
<tr>
<th></th>
<th>99% Credible intervals</th>
<th>0.005%</th>
<th>0.995%</th>
</tr>
</thead>
<tbody>
<tr>
<td>section 5.1</td>
<td>301677.2</td>
<td>309170.8</td>
<td></td>
</tr>
<tr>
<td>section 5.4</td>
<td>301748.2</td>
<td>309010.2</td>
<td></td>
</tr>
</tbody>
</table>

(Our latest credible interval is slightly tighter compared to the one in section 5.1.)
5.5 Visualization Results

Apart from the numerical results for our Superior Baselines, it is of great interest to investigate graphically the representation of them, by plotting the credible intervals in the same graph with the results from a 12-hour run. The following plots show two datasets (for the variables RrcConnEstabAtt and ErabRelAbnormalEnbAct) from a 12-hour run in comparison with the same Superior Baseline.

Plot 1. A lab run compared with the Superior Baseline for the variable RrcConnEstabAtt. The black dots represent measurements from a lab-run; the darker highlighted area is the 99% credible interval (Superior Baseline). X-axis: the dataset (lab-run), Y-axis: measurement period.
In addition, the calculated 99% quantiles for the observed lab-run in Plot 1 are (220609.5, 308900.5). It can be observed that the 99% quantiles are not included in the Superior Baseline intervals (301677.2, 309170.8) which were calculated in section 5.1 (the lower boundary of the observed lab-run is much smaller than the one in the Superior Baseline), as a result this will be assumed as a faulty run. This situation can clearly be captured graphically from plot 1. The same condition can be observed in plot 2, where the calculated 99% quantiles for the lab run are (4.00, 56.19) and the Superior Baseline intervals which were computed in section 5.2 are (0, 36).
6. Conclusions

To conclude, the mixture modeling procedure through the use of Gibbs Sampling, describes and fits pretty well our datasets. We categorized our datasets in three different cases, based on which distribution best fits to them and computed the Superior Baselines for each of our variables. Even after we changed the values for our hyper-parameters in the model, our credible intervals became barely tighter, something which in the case of datasets with large numbers is of minor concern. The same modeling process will be used in the computation of the Superior Baselines for the rest of the variables (1059) that exist in Ericsson’s database for each baseline run. Unfortunately, since this paper was produced under Ericsson’s guidance and support we are not allowed to publish our code and the datasets which were used.

A necessary question here is whether our modeling process is perfect or there are more ways which worth to be considered and investigated further. Of course there are a lot of different methods which can be used to upgrade our current modeling scheme. One alternative which has already been considered, concerns the datasets which are modeled through the procedure described in section 5.2, where we used estimates for the parameter \( r \) (number of failures) after fitting the Negative binomial distribution in our dataset. Instead, we can use the Metropolis-Hastings mechanism (see Frühwirth-Schnatter, 2006) for estimating \( r \).

In addition, the same approach (Metropolis-Hastings) can be used in order to investigate the behavior of our model in the case where we will assume that the number of components is unknown. In other words, estimate apart from the unknown parameters, the number of components for which we will have the best fitting. Meanwhile, of great concern would be the case to somehow specify covariates in our weights, which would be totally relevant with the characteristics of the lab sampling process, and use a mixture-of-experts model. A mixture-of-experts model is an extension of the finite mixture model to a regression setting (for more details and applications see Jacobs et al., 1991, Villani et al., 2009).

Moreover, another process which would be of interest to investigate further, although it concerns only the status in section 5.1, is called the Random Effects Model. The model idea is that we have for each level (baseline) a normal distribution with unknown mean and variance. Next, we assume that the four unknown means come from a normal distribution with some mean and a variance. Based on the random effects model we can estimate all the unknown parameters and make predictions for the mean values of each level, from which we can compute the prediction intervals for the measurements of future software runs (for more details see Montgomery, 2013, pp. 65-125).
References


Wikipedia, “Mixture Model”, en.wikipedia.org/wiki/Mixture_model (last accessed 03/05/2016).
Appendix
(histograms of the predictive distributions for the rest of the variables)

SessionTimeUe

CellHoPrepSuccLteIntraF
PdcpPktReceivedDl

Histogram of the dataset

Histogram of the predictive distribution

PdcpPktLostUl

Histogram of the dataset

Histogram of the predictive distribution
**SchedActivityCellUl**

Histogram of the dataset

Histogram of the predictive distribution

**RrcConnMax**

Histogram of the dataset

Histogram of the predictive distribution