An Infinite Multivariate Categorical Mixture Model for Self-Diagnosis of Telecommunication Networks

Amine Echraibi, Joachim Flocon-Cholet, Stéphane Gosselin, Sandrine Vaton

An Infinite Multivariate Categorical Mixture Model for Self-Diagnosis of Telecommunication Networks

Amine Echraibi, Joachim Flocon-Cholet, Stéphane Gosselin
Orange Labs
Lannion, France
{amine.echraibi, joachim.floconcholet, stephane.gosselin}@orange.com

Sandrine Vaton
IMT Atlantique
Brest, France
sandrine.vaton@imt-atlantique.fr

***Abstract*—The diagnosis of telecommunication networks remains a challenging task, mainly due to the large variety and volume of data from which the root causes have to be inferred. Expert systems, supervised machine learning, or Bayesian networks require expensive and time consuming data labeling or processing by experts. In this paper, we propose the Infinite Multivariate Categorical Mixture Model for clustering patterns of faults from data gathered from telecommunication networks. The model is able to automatically identify the number of clusters necessary to explain the data using the Dirichlet process prior. We show how to use Variational Inference to derive an Expectation-Maximization (EM) like algorithm to perform inference on the model. We apply our model on synthetic data generated from an expert Bayesian network of a Fiber-To-The-Home (FTTH) Gigabit capable Passive Optical Network (GPON). We show that the model discovers the patterns linked to the root causes of the faults with up to 96 % accuracy in an unsupervised manner. We also apply our method on real data gathered from the FTTH network and the local area network and demonstrate how the model is able to identify known faults.**

***Index Terms*—Infinite Mixture Models, Variational Inference, Self-Diagnosis, Access Network, Local Area Network, Pattern discovery**

I. Introduction

Identifying fault patterns in large telecommunication network and service infrastructures is a difficult task. Most of the considered technical solutions rely either on rule based expert systems or hand crafted expert Bayesian networks [1]–[4]. Although these approaches have had tremendous success, one of the unsung drawbacks is the data processing and the expert knowledge required to build the diagnosis model or rules. For expert systems, for example, the rules created by the expert require knowledge of the existing fault and the identification of the variables describing the fault. This process requires data processing by hand by an expert of the domain, which is an expensive and time consuming task. Also, the maintenance of the model or rules in the long run can be a significant issue for operational teams.

Recently, machine learning techniques have been tremendously successful in the identification and the extraction of patterns in various domains, such as text analysis, clustering documents and the identification of topics. In the context of our problem, similar approaches can be used to identify patterns of faults from diagnosis data. This task is thus an unsupervised machine learning task, where labels are not available and obtaining them is as complex as the data processing required to construct expert rules. However, unlike text data, the data gathered from various devices and services in the network is often structured in the form of a table, where each variable takes some range of values.

Clustering such data, gathered from telecommunication networks and services presents many challenges. The first and the main challenge is the unknown number of clusters of faults in the data. The second challenge is the types and multivariate nature of the data. The data is multi-dimensional and can contain categorical and continuous variables. Therefore, classical clustering algorithms where the number of clusters is to be set a priori require some form of model selection. Furthermore, classical approaches such as KMeans suppose a specific probability distribution for each cluster, notably Gaussian distributions with a diagonal constant covariance matrix and uniformly distributed mixture weights. These modeling assumptions can hurt the performance of the clustering when the data do not comply with such assumptions, which is often the case when dealing with real-world applications.

In this paper, we propose an infinite multivariate categorical mixture model to identify patterns of faults in an unsupervised setting, without any prior expert knowledge. The model is based on the Dirichlet Process prior introduced in [5], which allows for learning the number of clusters from the data itself. However, the Dirichlet Process supposes an infinite number of clusters which translates to a harder intractable inference problem on the model. Our contributions are the following:

- We provide a theoretical formulation of the infinite multivariate categorical mixture model (section 2).
- We show how to perform approximate inference on the model, in order to extract the clusters from the data using Variational Inference [6] (section 3).
- We demonstrate how the model is able to identify root causes of faults in a synthetic dataset generated from a real-world expert Bayesian Network (section 4).
- We also demonstrate the clustering performance of the model on real operational data acquired from the Fixed Access Network and the Local Area Network (section 5).

Implementation of the model and synthetic data are available in: https://git.io/JejBQ

978-1-7281-5127-4/20/$31.00 ©2020 IEEE

II. The Infinite Categorical Mixture Model

A. Notations

We introduce some notations that we will use throughout the paper. We denote by X_i an observable random variable, describing a specific feature of a network equipment such as a status, an alarm or a physical metric. We consider only categorical random variables. Continuous variables such as optical powers or temperatures are discretized, using standard methods such as equal frequency or equal width discretization. Thus, each random variable takes values in $\text{Val}(X_i) = \{v_{1i}, ..., v_{|X_i|i}\}$, where $|X_i|$ is the number of modalities of variable X_i. Let $x_{1:N}$ represent N samples of the vector $X = [X_1, ..., X_d]^T$ of dimension d. We denote by $z_{1:N}$ N random variables, where z_n represents the diagnosis cluster of sample x_n. The Kullback-Leibler divergence between two distributions q and p is denoted by:

$$\mathbb{D}_{KL}\left[q||p\right] = \int q(x) \log \frac{q(x)}{p(x)} dx$$

The entropy of a distribution p is denoted by:

$$\mathbb{H}\left[p\right] = -\int p(x) \log p(x) dx$$

For discrete radom variables integrals are replaced by discrete sums over the values taking by the random variable X. We denote by $\text{Cat}(x|\pi)$ the categorical distribution of a random variable X taking discrete values $\{x_1, ..., x_m\}$ with probabilities $\{\pi_1, ..., \pi_m\}$:

$$\text{Cat}(x|\pi) = \prod_{i=1}^{m} \pi_i^{\mathbb{1}[x=x_i]} \quad \text{s.t} \quad \sum_i \pi_i = 1$$

We denote by $\text{Beta}(\beta; 1, \eta)$ the beta distribution of parameters 1 and η, defined as:

$$\text{Beta}(\beta; 1, \eta) = \text{C}(1-\beta)^{\eta-1} \quad \text{C: normalizing constant}$$

If a random variable X has a probability distribution $p_{_X}(x)$, we simply write:

$$x \sim p(\cdot)$$

We denote by $x_{1:m}$ the vector of elements $\{x_1, ..., x_m\}$ and x_{-i} the vector of all elements except the i^{th} index.

B. The Dirichlet Process Prior

In a model-based clustering, one of the main challenges is to choose the correct number of clusters to analyze the data. In our case, the number of diagnosis clusters is unknown a priori. The Dirichlet Process (DP) [5] allows us to introduce a prior on the number of clusters, without fixing it explicitly. One way to construct the Dirichlet Process is via the stick-breaking construction [7], where η is the concentration parameter of the DP ($\eta > 0$) , and the weight π_k of the k^{th} cluster is constructed from k samples drawn from a beta distribution as follows:

$$\begin{aligned} \beta_k &\sim \text{Beta}(\cdot; 1, \eta) \\ \pi_k &= \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \end{aligned}$$

The stick breaking construction is as follows, π_k represents the length of the k^{th} piece broken from a stick of length 1. If the concentration parameter is small, π_k will be large (close to 1) at the first times the stick is broken, therefore the stick will be broken a small number of times (small number of clusters). If η is large, π_k will be small (close to 0) and the stick can be broken a large number of times (large number of clusters). The probability that the $(n+1)^{\text{th}}$ data point belongs to a new cluster k^* or K existing clusters is [8]:

$$\mathbb{P}[z_{n+1} = z | z_{1:n}, \eta] = \frac{1}{\eta + n} \left[\eta \mathbb{1}[z = k^*] + \sum_{k=1}^{K} n_k \mathbb{1}[z = k] \right]$$

where n_k is the number of data points in cluster k. If $\eta \to \infty$ a new cluster would be created for each data point, and if $\eta \to 0$ all data points are concentrated in the first cluster. The intuition behind the Dirichlet process is the following: as the number of data points increases, we allow the number of clusters to grow according to the concentration parameter and the clusters already assigned. New samples are assigned to existing clusters if they match, otherwise a new cluster is created for them. Therefore the Dirichlet Process allows the mixture model to cluster the data and identify automatically the number of clusters necessary to explain the data.

C. The Dirichlet Process Categorical Mixture Model (DPCMM)

Using the stick-breaking construction introduced in the previous section, the generative process for the model becomes:

$$\begin{aligned} \beta_k &\sim \text{Beta}(\cdot; 1, \eta) \\ \pi_k &= \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \\ z_n &\sim \text{Cat}(\cdot|\pi) = \prod_{k=1}^{\infty} \pi_k^{\mathbb{1}[z_n = k]} \\ b_{ki} &\sim \text{Dir}(\cdot; \alpha_i, |X_i|) = \prod_{v \in Val(X_i)} b_{k,i,v}^{\alpha_{iv} - 1} \\ &\quad \text{s.t} \sum_{v \in Val(X_i)} b_{k,i,v} = 1 \\ x_{ni} | z_n = k, b &\sim \text{Cat}(\cdot|b_{ki}) = \prod_{v \in Val(X_i)} b_{k,i,v}^{\mathbb{1}[x_{ni} = v]} \end{aligned}$$

Figure 1 shows a graphical representation of the generative process of the model.

Based on the assignment of cluster z_n a sample x_{ni} is drawn based on the conditional probability distribution for variable X_i taking a certain value v under cluster $z_n = k$:

$$b_{k,i,v} = \mathbb{P}[x_{ni} = v | z_n = k]$$

In the classical Bayesian formulation considered above, the parameters $b_{k,i} = [b_{k,i,v}]_{v \in Val(X_i)}$ are themselves random variables with a conjugate prior, in this case a Dirichlet distribution with concentration parameter α_i for variable X_i. The concentration parameter allows us to inject prior

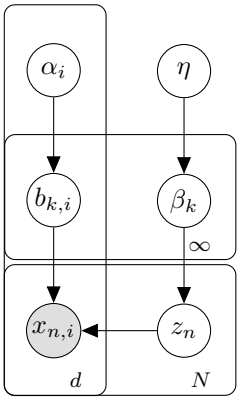

Fig. 1. Graphical representation of the model in plate notation.

knowledge about the modalities of variable X_i. α_{iv} thus represents a weight for modality v in the Dirichlet distribution associated with variable X_i. In the case where no information is available we use an uninformative prior $\alpha_{iv} \propto \frac{1}{|X_i|}$.

In order to fit the model to the data and identify the clusters representing the different types of network failures, we need to compute or approximate the posterior distribution:

$$p(z_{1:N}, b, \beta | x_{1:N}) = \frac{p(z_{1:N}, b, \beta, x_{1:N})}{p(x_{1:N})} \quad (1)$$

Markov Chain Monte Carlo (MCMC) approaches are commonly used to fit such models [9]. The main idea behind these approaches is to run a Markov chain long enough (until convergence) where the stationary distribution at convergence is the posterior of interest as defined in equation (1). One major drawback of such approaches is the long burn-in time of the Markov chain that does not scale well with the number of dimensions of the data [10]. In a high dimensional space the Markov chain needs to visit a high number of states. The other drawback is that convergence of MCMC methods is hard to diagnose, due to the sampling issue in high dimensions. Although MCMC methods have had many successes in small scale problems, further research is required particularly for large scale applications.

Recently, an alternative approach emerged called Variational Inference [6], [8], [10]. The main idea behind the approach is to express the intractable inference problem as a relaxed optimization problem, where we can leverage all the tools available in the mathematical literature of optimization to solve the inference problem. In the next section, we present a brief review of variational inference, and we show how we can use it to approximate our posterior distribution of equation (1).

III. Variational Inference

A. Variational Inference and the mean field approximation

In order to introduce the variational inference approach, we denote by $\zeta_{1:p} = \{z_{1;N}, b, \beta\}$ the vector grouping all the hidden variables of the model. Solving the inference problem of the model amounts to determining $p(\zeta_{1:p}|x_{1:N})$. As mentioned previously, close form solutions for this quantity can not be determined. Variational inference and the mean-field approximation allow us to approximate this intractable distribution by a set of distributions for which the inference is tractable, namely the mean field family. A distribution q is said to be in the mean field family for variables $\zeta_{1:p}$, if it verifies:

$$q(\zeta_{1:p}) = \prod_{l=1}^{p} q(\zeta_p)$$

The main idea of variational inference is to approximate the intractable distribution (1), by finding the closest mean field family member in terms of Kullback-Leibler divergence i.e:

$$q^* = \min_q \mathbb{D}_{KL}\left[q(\zeta_{1:p}) || p(\zeta_{1:p}|x_{1:N})\right]$$

By exploiting the factorization in the mean field family, we can show that the solution q^* verifies the following fixed point equations:

$$\log q^*(\zeta_l) = \text{const} + \mathbb{E}_{\zeta_{-l} \sim q^*}\left[\log p(\zeta_{1:p}, x_{1:N})\right] \quad \forall l \quad (2)$$

For an explicit derivation of this criterion we refer the reader to [8]. In the case of our model, the mean-field family is defined as :

$$q(z_{1:N}, b, \beta) = \prod_{n=1}^{N} q(z_n) \prod_{i=1}^{d} \prod_{k=1}^{T} q(b_{k,i}) \prod_{k=1}^{T} q(\beta_k) \quad (3)$$

We also suppose that $q(\beta_T = 1) = 1$ hence $q(z_n > T) = 0$, i.e the number of clusters is truncated to an upper bound on the true number of clusters [10]. A noteworthy aspect of this approach is that the true posterior given by (1) has an infinite number of factors, however the approximating distribution q is constrained based on the previous conditions. Therefore, the true model is unchanged however the minimization problem is relaxed in order to be solved efficiently.

B. Variational Inference for The DPCMM

By applying equation (2) to our Infinite Categorical Mixture Model, we obtain:

$$\begin{aligned}
\log q^*(z_n) &= \text{const} + \mathbb{E}_{\{z_{-n}, \beta, b\} \sim q^*}\left[\log p(z_{1:N}, b, \beta, x_{1:N})\right] \\
\log q^*(b_{k,i}) &= \text{const} + \mathbb{E}_{\{z_{1:N}, \beta, b_{-\{k,i\}}\} \sim q^*}\left[\log p(z_{1:N}, b, \beta, x_{1:N})\right] \\
\log q^*(\beta_k) &= \text{const} + \mathbb{E}_{\{z_{1:N}, \beta_{-k}, b\} \sim q^*}\left[\log p(z_{1:N}, b, \beta, x_{1:N})\right]
\end{aligned}$$

And by substituting the expression of $p(z_{1:N}, b, \beta, x_{1:N})$ resulting from the graphical representation of the model (Figure 1), we then deduce the following approximating distributions and fixed point updates for their parameters:

$$\begin{aligned}
q^*(z_n) &= \text{Cat}(z_n; \phi_n) \\
q^*(b_{k,i}) &= \text{Dir}(b_{k,i}; \epsilon_{k,i}, |X_i|) \\
q^*(\beta_k) &= \text{Beta}(\beta_k; \gamma_{1,k}, \gamma_{2,k})
\end{aligned}$$

The mean field fixed point equations for the parameters are finally the following, where ψ is the digamma function:

$$\begin{aligned}\log \phi_{nk} &= \text{const} + \sum_{i=1}^{d} \sum_{v \in Val(X_i)} \mathbb{1}[x_{ni} = v][\psi(\epsilon_{k,i,v}) \\ &- \psi(\sum_{v' \in Val(X_i)} \epsilon_{k,i,v'})] \\ &+ \psi(\gamma_{1,k}) - \psi(\gamma_{1,k} + \gamma_{2,k}) \qquad (4) \\ &+ \sum_{l=1}^{k-1} [\psi(\gamma_{2,l}) - \psi(\gamma_{1,l} + \gamma_{2,l})] \quad \text{s.t} \quad \sum_{k=1}^{T} \phi_{nk} = 1 \end{aligned}$$

$$\gamma_{1,k} = 1 + \sum_{n=1}^{N} \phi_{nk} \qquad (5)$$

$$\gamma_{2,k} = \eta + \sum_{n=1}^{N} \sum_{l=k+1}^{T} \phi_{nl} \qquad (6)$$

$$\epsilon_{k,i,v} = \alpha_{i,v} + \sum_{n=1}^{N} \phi_{nk} \mathbb{1}[x_{ni} = v] \qquad (7)$$

C. Convergence and an Algorithm

In order to monitor convergence while iterating the fixed point equations, we can plot the evidence lower bound defined as:

$$\begin{aligned}\mathcal{L}(q) &= -\mathbb{D}_{KL}\left[q||p(z_{1:N}, b, \beta, x_{1:N})\right] \qquad (8) \\ &= \mathbb{E}_{\{z_{1:N}, \beta, b\} \sim q}\left[\log p(z_{1:N}, b, \beta, x_{1:N})\right] + \mathbb{H}[q]\end{aligned}$$

The mean field fixed point updates minimize $-\mathcal{L}(q)$, therefore across iterations the evidence lower bound should increase monotonically. Usually, we only evaluate the log predictive across iterations, which is the first term of $\mathcal{L}(q)$. This quantity is not guaranteed to increase monotonically. However, at convergence, this quantity reaches a plateau, and by evaluating convergence in this manner we can bypass the tedious calculation of different entropy terms in $\mathbb{H}[q]$.

Like the EM algorithm, the fixed point update equations only guarantee convergence to a local minimum depending on the initialization. Therefore, in order to converge to the best local minimum possible we need to initialize the parameters ϕ_n in the best way possible. One approach widely used for the initialization of mixture models is to initialize ϕ_n from a KMeans algorithm. Hence, given the centers of a fitted KMeans μ_k where we set the number of clusters to the upper bound T, we initialize ϕ_n as:

$$\phi_{nk} \propto \exp\left(-\frac{1}{2}||x_n - \mu_k||\right)$$

Algorithm 1 presents the inference process on the Dirichlet Process Categorical Mixture Model using variational inference. Similar to the EM Algorithm, we can recognize two steps in the iterative process: the update of the local variational parameters ϕ_n analogous to the E-step, and the update of the global variational parameters γ_1, γ_2, and ϵ analogous to the M-step.

Algorithm 1 Variational Inference for the DPCMM

Input: $x_{1:N}, T, \eta$
$\{\mu_k\}_k = \text{KMeans}(T, x_{1:N})$
$\phi_{nk}^{(0)} \propto \exp\left(-\frac{1}{2}||x_n - \mu_k||\right)$ {Initialize ϕ_n $\forall k, \forall n$}
$\mathcal{L}^{(0)} = -\infty$
for $t = 1...\infty$ **do**
  Compute: $\gamma_{1,k}^{(t)}$ $\forall k$ (5)
  Compute: $\gamma_{2,k}^{(t)}$ $\forall k$ (6)
  Compute: $\epsilon_{k,i,v}^{(t)}$ $\forall v, \forall k, \forall i$ (7)
  Compute: $\phi_{n,k}^{(t)}$ $\forall n, \forall k$ (4)
  Compute $\mathcal{L}^{(t)}$ (8)
  if $|\frac{\mathcal{L}^{(t)} - \mathcal{L}^{(t-1)}}{\mathcal{L}^{(t)}}| \leq 10^{-6}$ **then**
    break
  end if
end for
$z_n = \arg\max_k \phi_{nk}$ $\forall n$
return $z_{1:N}, \phi$

IV. Assessment on Synthetic Data Generated by an Expert Bayesian Network for FTTH Diagnosis

A. Experiment and Dataset

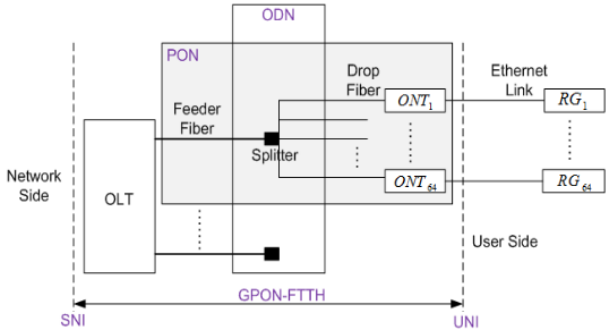

Fig. 3. FTTH GPON Architecture [1].

In this first experiment we assess our clustering model for the FTTH fault diagnosis. We reuse the work done by Tembo et al. [1] where the authors designed an expert Bayesian network that reflects the behaviour of a real GPON network (Figure 3). The Bayesian network is depicted on Figure 2 where the colored top nodes represent the root causes and the other nodes the observations coming from the network. The goal here is to see if our clustering model is able to automatically detect the patterns corresponding to the different root causes.

In order to generate the synthetic data for the experiment, we simulate a fault by activating the root cause for the fault, and we sample the visible variables. Here we sampled 6 uncorrelated faults (for more details on the expert bayesian network and remaining variables, we refer the reader to [1]):

- **AltONT:** highlights a problem with the power supply of the optical network termination (ONT). This hidden

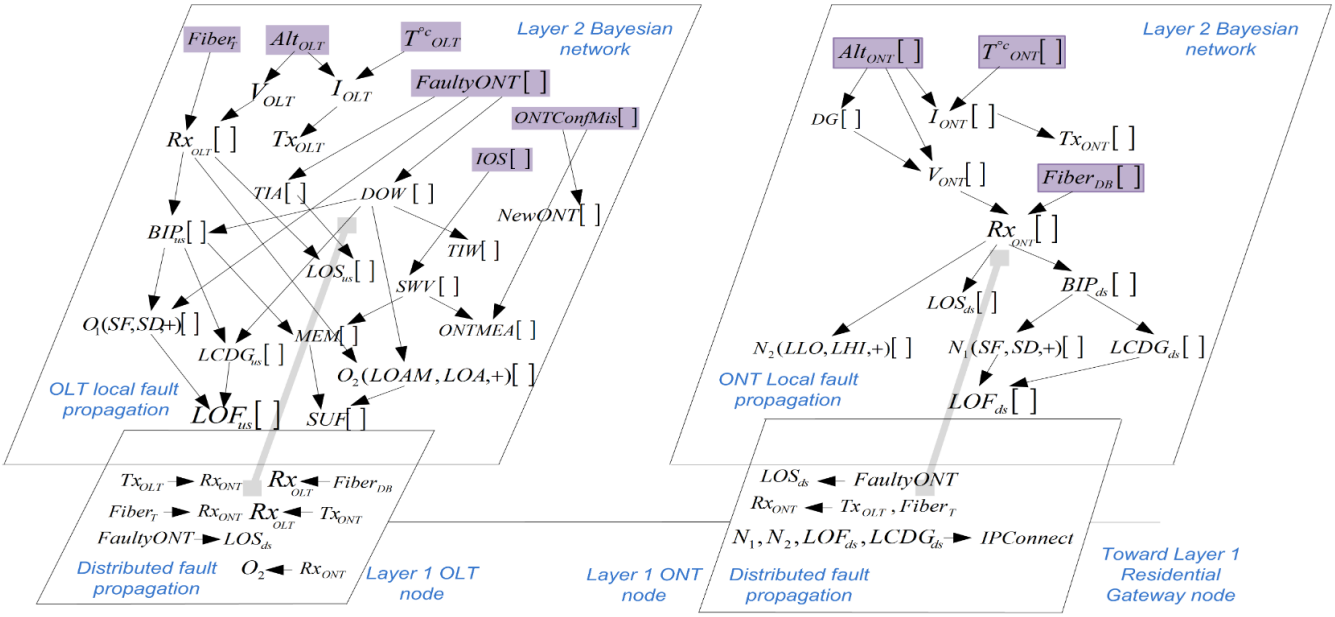

Fig. 2. Expert Bayesian Network of the FTTH GPON [1]. Colored nodes are possible root causes whereas non-colored nodes are observations.

variable controls the current of the ONT I_{ONT}, the Dying Gasp alarm DG, and the electric voltage of the ONT V_{ONT}.

- **AltOLT:** describes a problem with the power supply of the optical line termination (OLT). It controls similar variables for the OLT, V_{OLT} and I_{OLT}.
- **FaultyONT:** denotes whether the ONT is faulty, and symbolizes the global state for an ONT. It controls the alarms TIA (Transmission Interference Alarm), DOW (Drift Of Windows), and SF (Signal Fail) or SD (Signal Degraded).
- **FiberDB:** represents the state of the drop optical fiber and controls Rx_{ONT} the power at which the ONT receives the signal.
- **IOS:** (Image Operating System) refers to an incompatibility between the OS of an OLT and the OS of an ONT. It controls the variable SWV describing the software version alarm.
- **TcOLT:** represents the temperature of the OLT.

We sampled 150 data points for each fault, using the likelihood weighted sampling method, based on the conditional distribution tables of each observation given the root cause. Therefore we generate for each fault a pattern of the visible variables for a specific customer equipment (ONT). The resulting dataset contains 900 samples, where each sample presents a realisation of the 29 visible nodes of the Bayesian network. These visible nodes are referred to in our modeling by X_i.

B. Evaluation Process and Results

In order to demonstrate the value of the Dirichlet Process prior, we suppose that the number of faults is unknown, similarly to many real-world applications. As discussed in section III-A we evaluate our model with a truncation level $T = 50$ which represents an upper bound of the true number of clusters (here $K = 6$). The algorithm will build up to 50 clusters but will automatically keep the main relevant clusters. We set the concentration parameter of the Dirichlet process to $\eta = 0.001$. We run our model on the generated dataset multiple times and we plot the evidence lower bound defined in equation (8) in Figure 4.

The evaluation metric is the clustering accuracy. It is similar to the classification accuracy, however, in the clustering task the clusters are not identifiable with the labels, i.e the clusters can change from one run of the algorithm to another. Therefore we need to test all possible combinations and choose the best one. The clustering accuracy is defined as [11]:

$$\text{ACC} = \max_{m \in \mathcal{M}} \frac{\sum_{n=1}^{N} \mathbb{1}\left[l_n = m(z_n)\right]}{N}$$

where z_n is the cluster assignment, l_n the true label and $\mathcal{M}$ the set of all possible one-to-one mappings.

The best run of the model was selected as the run with highest evidence lower bound (Experiment 6). It corresponds to a clustering accuracy of **0.96** as seen on the table I. We report the confusion matrix between the clusters and true labels in table II.

The confusion matrix in table II shows that each of the main clusters found by the clustering model corresponds clearly to

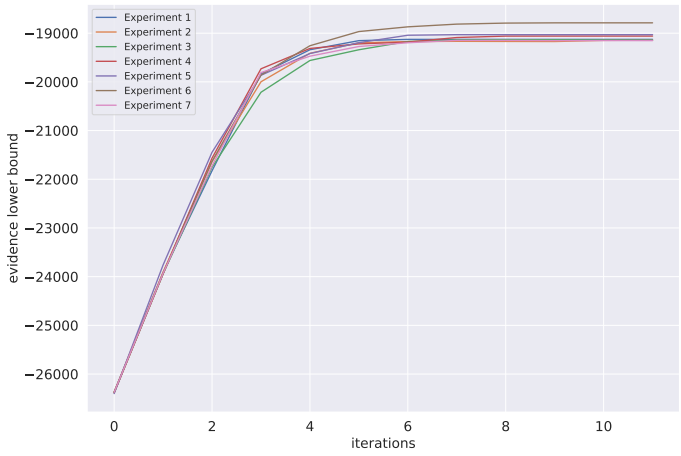

Fig. 4. The evidence lower bound for different runs of the algorithm.

Best model (highest $\mathcal{L}$)	Exp. 6
ACC	**0.96**

TABLE I
CLUSTERING ACCURACY FOR THE BEST CLUSTERING MODEL.

a particular root cause, with up to 3.2 % error. The Dirichlet process automatically identifies the main patterns and clusters them in 6 clusters corresponding to the true clusters. The other clusters remain empty. It shows that the number of clusters is automatically and solely estimated from the data. Here, based on a ground truth we can also assess the relevancy of each cluster since they refer to a particular root cause.

V. Extraction of Patterns of Faults in The Fixed Access Network and the Local Area Network

In this second experiment, we apply our clustering model on real operational network data. However in this setup we don't have access to a ground truth so the aim of this experiment is to demonstrate the benefit of our clustering model in a data exploration process.

A. Data Collection and Network Scope

We place our working environment on the FTTH GPON fixed access network and the local area network as represented in Figure 5. For the fixed access network we collected data describing the status of the OLT, and the status of the ONT (olt_status, ont_status, ont_download_status). For the local area network, we collected data describing the status of the router and the different services provided such as, IPTV, VOD (Video On Demand) and VOIP. We also collected data describing the account status of the client. The final dataset contains about 7000 clients who claimed to have a problem with one of their services. For each client we collect 29 variables reflecting the different aspects of the environment as described previously.

	AltOLT	AltOLT	FaultyONT	FiberDB	IOS	TcOLT
cluster 11	**150**	0	0	0	0	0
cluster 25	0	**132**	0	0	0	0
cluster 5	0	0	**148**	0	0	0
cluster 3	0	0	1	**150**	0	2
cluster 39	0	1	0	0	**147**	3
cluster 17	0	17	1	0	3	**145**
cluster 1	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
cluster 50	0	0	0	0	0	0

TABLE II
CONFUSION MATRIX.

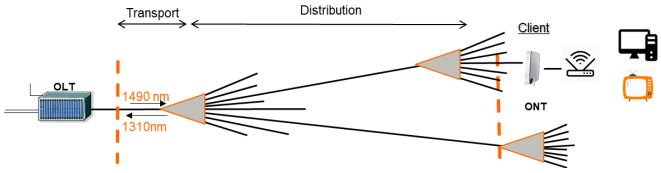

Fig. 5. Network scope for fault pattern extraction from real operational data

B. Cluster Analysis

We apply the same protocol as the previous experiment described in section IV-B. We obtained 6 main clusters corresponding to known problems. We report for each cluster the number of customers assigned to the cluster and the most common pattern, i.e. characteristic values that the variables take and the counts for each variable (Figure 6). Furthermore, we give an interpretation based on network expertise to understand what kind of problem the customers are facing in each cluster.

- **Cluster 37**: (600 clients) This cluster gathers customers with a problem on the remote PVR for the IPTV: Remote_PVR = Missing, and all other values correspond to an operational state of the client's line: (client_account_status=1, OLT_status=OK, router_status=Enabled, ONT_status=NODEFECT...).
- **Cluster 30:** (> 2500 clients) In this cluster the ONT is not detected: (ONT_status = Missing), where all other services are operational (router_status=Enabled, voip_status=Ok, tv_profile_status=operational ...).
- **Cluster 43:** (≈ 2500 clients) This cluster corresponds to the optimal behavior of the network and services, all equipment are detected and all services are operational.
- **cluster 44:** (≈ 550 clients) This cluster describes a problem with the router where the router is disabled (router_status=DISABLED, router_ipv6_status=DISABLED), however all other equipments are operational and client account status is activated.
- **Cluster 40:** (≈ 27 clients) This cluster represents the case where an account is suspended and the VOIP service is down (client_account_status = 0, tv_profile_status=SUSPENDED, VOIP_status=KO), however all pieces of equipment are operational.
- **Cluster 12:** (≈ 130 clients) This cluster gathers clients

with deactivated WiFi (router_WiFi_status = Down, router_status=Enabled), and all other equipment is operational (ont_status = NODEFECT, OLT_status=OK...).

This experiment shows that our clustering method can be helpful in a data exploration process where the number of clusters is not known a priori. With a simple network expertise we can see that the clusters are relevant and describe a particular behaviour on the network. We can also stress the fact that this method can discover the main patterns in the data but also patterns that are not dominant but still relevant. For example, we have clusters composed of 2500 customers (cluster 43) and clusters with only dozens of customers (cluster 40). This is thus an interesting feature since we can discover "weak signal" patterns.

VI. Discussion and Related works

Machine learning techniques for fault detection in telecommunication networks are promising. Bayesian networks have been dominant in this domain [1]–[4]. The main advantage of Bayesian network modeling is the easy interpretable nature of the approach. Root causes of faults are explicitly modeled by variables of the Bayesian network and inference can be performed in order to determine the root cause from the evidence presented by the observable variables. This approach however, as previously mentioned, requires expert knowledge of the specific domain of faults and a time consuming task to build the Bayesian network and all the dependencies between the variables. Recently, researchers have been interested in learning the structure of the Bayesian network from data [12], and such approaches have been successfully used for the self-diagnosis problem [13]. However, the main problem, in such approaches, is the loss of the interpretable nature of Bayesian networks. The structure learned from the data can be one of a class of equivalence of optimal structures, hence, this often results in structures that are optimal but hard to interpret. The second group of machine learning approaches, are anomaly detection based approaches [14], [15]. Although these methods allow for accurate detection of anomalies from data, their main drawback is that all anomalies identified are grouped in a single cluster. Classifying each anomaly requires either a relabeling process by hand or a clustering process. Therefore, the natural reformulation of the self-diagnosis problem is as a clustering problem. Clustering types of faults from the data has been previously proposed [16], [17]. However, most of these approaches rely on classical algorithms where the number of clusters is known a priori or estimated by model selection. Although our approach allows for automatic determination of the number of clusters and requires no intervention from an expert, one drawback is the interpretability of the clusters. A post processing of the clusters is needed in order to identify the fault present in each cluster and the root cause. [18] proposed a method based on decision trees to accomplish this task.

VII. Conclusion and Future works

In this paper, we showed how the infinite multivariate categorical mixture model can be used to analyze data gathered from telecommunication networks and services and how to discover fault patterns in an unsupervised manner. The model is capable of identifying the correct number of clusters to analyze the data using the Dirichlet process. We showed how Variational Inference can be used to perform inference on the model. This approach allows inference to scale well with the dimensionality of the data, and the convergence of the model can be determined explicitly. In future work, we will explore how such methods can be applied to time series data. Analysis of such data gathered from telecommunication networks can also be very useful for anomaly detection and monitoring.

References

[1] S. R. Tembo, S. Vaton, J. L. Courant, S. Gosselin, and M. Beuvelot, "Model-based probabilistic reasoning for self-diagnosis of telecommunication networks: Application to a gpon-ftth access network," *Journal of Network and Systems Management*, vol. 25, no. 3, pp. 558–590, Jul 2017.

[2] P. Kogeda and J. I. Agbinya, "Prediction of faults in cellular networks using bayesian network model," in *International conference on Wireless Broadband and Ultra Wideband Communication*. UTS ePress, 2006.

[3] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper, "The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks," in *AIME 89*. Springer, 1989, pp. 247–256.

[4] O. P. Kogeda, J. I. Agbinya, and C. W. Omlin, "A probabilistic approach to faults prediction in cellular networks," in *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)*. IEEE, 2006, pp. 130–130.

[5] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230, 1973.

[6] M. J. Wainwright, M. I. Jordan *et al.*, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[7] J. Sethuraman, "A constructive definition of dirichlet priors," *Statistica sinica*, pp. 639–650, 1994.

[8] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[9] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of computational and graphical statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[10] D. M. Blei, M. I. Jordan *et al.*, "Variational inference for dirichlet process mixtures," *Bayesian analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[11] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.

[12] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[13] L. Bennacer, Y. Amirat, A. Chibani, A. Mellouk, and L. Ciavaglia, "Self-diagnosis technique for virtual private networks combining bayesian networks and case-based reasoning," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 354–366, 2014.

[14] R. A. Maxion, "Anomaly detection for diagnosis," in *[1990] Digest of Papers. Fault-Tolerant Computing: 20th International Symposium*. IEEE, 1990, pp. 20–27.

[15] C. S. Hood and C. Ji, "Proactive network-fault detection [telecommunications]," *IEEE Transactions on reliability*, vol. 46, no. 3, pp. 333–341, 1997.

[16] U. S. Hashmi, A. Darbandi, and A. Imran, "Enabling proactive self-healing by data mining network failure logs," in *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2017, pp. 511–517.

[17] M. Adda, K. Qader, and M. Al-Kasassbeh, "Comparative analysis of clustering techniques in network traffic faults classification," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 4, pp. 6551–6563, 2017.

[18] M. Chen, A. X. Zheng, J. Lloyd, M. I. Jordan, and E. Brewer, "Failure diagnosis using decision trees," in *International Conference on Autonomic Computing, 2004. Proceedings.* IEEE, 2004, pp. 36–43.

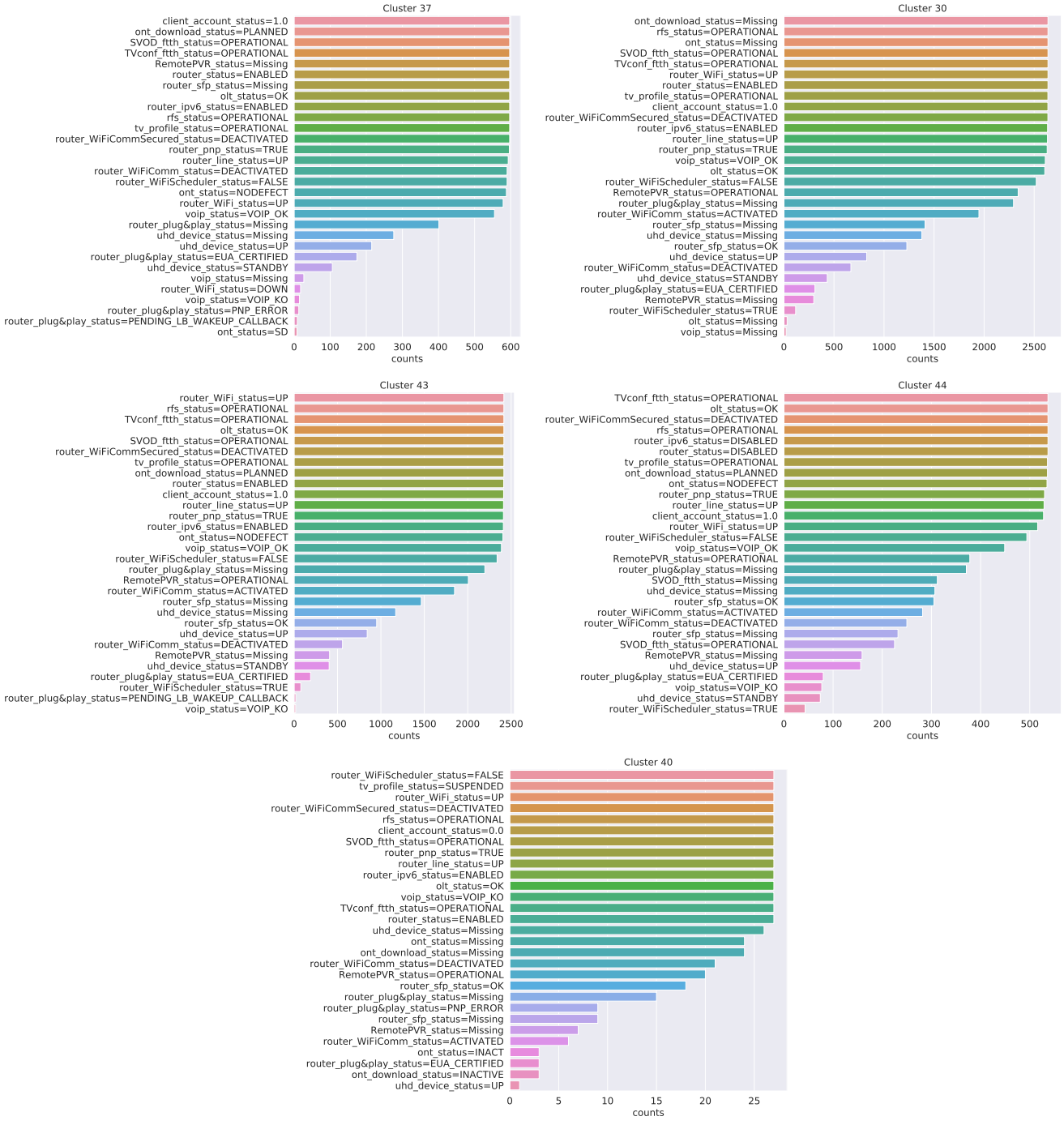

Fig. 6. Assignments defining each cluster from most occurring to least occurring