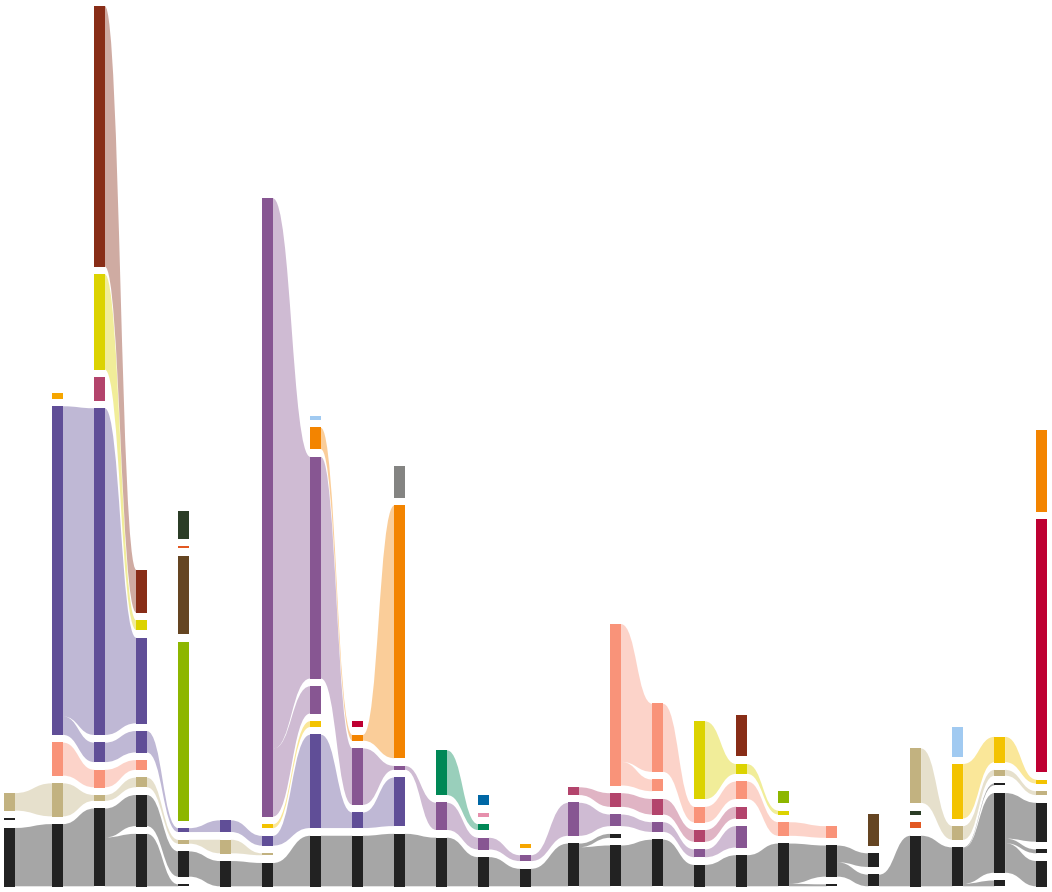


Nonparametric Evolutionary Short Text Topic Modeling

Master's Thesis in Complex Adaptive Systems

EMIL EJBYFELDT

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Master's Thesis 2018:DATX05

Nonparametric Evolutionary Short Text Topic Modeling

Emil Ejbyfeldt

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Nonparametric Evolutionary Short Text Topic Modeling
Emil Ejbyfeldt[1]

Email:
[1]emil.ejbyfeldt@gmail.com

Master thesis at the Department of Computer Science and Engineering, Chalmers

Advisor: Staffan Truvé, Recorded Future
Supervisor: Morteza Chehreghani
Examiner: Devdatt Dubhashi

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
SE-412 96 Gothenburg
Sweden

Cover:
Visualization of topic evolution (bar are topics and their height represents the number of topics and the Bezier curves connecting topics represent that the topic on the right topic inherits from the other), see Chapter 4.

Abstract

With the advent of social media more information is published and discussions happens in the form of short text. Tools are needed for detecting new and changes in topics that can help people understand and explore the vast amount of information available. Many of current approaches do not handle short text well and some require specification of the number of topics beforehand. A way of extending Dirichlet Processes Mixture Models to handle temporal data is introduced. A collapsed Gibbs sampling algorithm for interference is derived for the model. In the model data is divided into epochs where data is interchangeable within an epoch. The number of clusters in each epoch is unbounded and the model has the ability to recover the birth, death and split of clusters. Topic modeling is done by assuming that each short text belong to a single topic. The model is specifically evaluated on short text dataset to show the model's ability to discover topic evolution and discover the appearance of new topics. We also show that the model has better stability and less overfitting than previous solutions with the same abilities.

Acknowledgements

I would like to thank Recorded Future for providing a place to work and coffee throughout the project. Specifically, I would like to thank Staffan Truvé for being my supervisor at Recorded Future and providing many helpful suggestions and ideas. I would also like to thank Alexander Karlsson at Högskolan i Skövde who provided me with an introduction to probabilistic topic models and provided insightful discussions during the project. Lastly, I would like to thank my supervisor Morteza Chehreghani and examiner Devdatt Dubhashi at Chalmers.

Emil Ejbyfeldt, Gothenburg, May 2018

Contents

Chapter 1

Introduction

Increasingly, information published online comes in the form of short text, for example, feeds, tweets, forums, and status messages. New information often quickly reaches these types of media and therefore detecting trends in topics is of interest. The large amount of information has created a need for methods of exploration, organization, and classification. The popular methods used for topic modeling for longer documents, such as Latent Dirichlet Allocation (LDA) [1] do not perform well on the shorter texts due to the low word count and subsequent data sparsity. Thus, specialized methods are necessary for evolutionary topic modeling on short texts.

In LDA documents are modeled using a "bag-of-words" model where a document is only represented by the frequency of word occurrences. Each document is then assumed to be a mixture of topics and each topic is assumed to be a distribution of words. Due to the high order of the model it is not suitable for short texts with low word count. A simpler statistical model with a similar structure for short text was proposed in [2]. In this model, each document is instead assumed to only come from a single topic. This model was shown to work well for shorter texts.

Both LDA and the model used in [2] require specifying the number of topics, which for a new dataset is intractable in general. It is possible to generalize both these methods using Dirichlet processes (LDA requires the use of hierarchal Dirichlet processes) to models that do not require specification of the number of topics in advance [3]. These generalizations are examples of Bayesian nonparametrics [3] and are useful when the number of topics is not known in advance.

In order to find the topic assignments in LDA, it is required to calculate the posterior distribution of the hidden variable in the model. Calculating the exact distribution is intractable, so instead an approximation inference scheme is usually applied. One approximate method is Gibbs sampling in the context of a Markov Chain Monte Carlo algorithm. How to develop a Gibbs sampling for a Dirichlet process mixture model is described in [4].

This thesis continues the work of developing short texts topic modeling. An evolutionary temporal extension for Dirichlet processes is introduced. This extension is used to create a nonparametric evolutionary short text topic model where each document is modeled as a having a single topic. The topic model is evaluated on real datasets.

1.1 Purpose and Aim

Recorded Future harvests and analyzes tens of millions of documents per day, and performs natural language processing (NLP) to turn unstructured text into structured data. Since the volumes are so big, and since there are many copies and versions of each story, they have a great need to cluster different text elements (fragments/event references) together, both to find the most important topics, and to track trending topics over time.

The aim of the project is to evaluate evolutionary topic modeling for short texts. Specifically to implement an algorithm based on Bayesian statistics and Dirichlet processes for that purpose. The algorithm will be evaluated empirically on available datasets.

1.2 Structure of the thesis

Chapter 2 introduces the theoretical background for the thesis. The chapter first introduces the background to computational Bayesian inference. Then Dirichlet processes and their use in non-parametric Bayesian topic modeling are discussed. In Chapter 3 we introduce our evolutionary extension of a Dirichlet process mixture model and derive a Gibbs sampling algorithm for inference. Results from using the model on real datasets are presented in Chapter 4, and the results are given a more general discussion in Chapter 5. The thesis is concluded in Chapter 6 with a summary and ideas for future research.

Chapter 2

Background

In this chapter we introduce the theoretical background for probabilistic topic models based on Dirichlet processes. Firstly, we introduce the bag of words model and how it is used in topic modeling. Then we give an explanation of Bayesian inference and how it can be done computationally using Gibbs sampling. Next, Dirichlet processes and how they are used for infinite mixture models and nonparametric topic modeling is introduced. The chapter is concluded with explanation of perplexity and how it is used for evaluation.

2.1 Bag of words

The bag of words is a simplified representation of text. All words in a document are assumed to be exchangeable and therefore, the document can be represented by the number of times each word appears. If a fixed vocabulary is used then the document can instead be represented by a word vector. Each element i is number of times a word, with index i in the vocabulary, appears in the document. This representation is commonly used in probabilistic topic models [1, 2, 5]. In the case of [2] the word vector is modeled as a single draw from a multinomial distribution. However for LDA [1] or DTM [5] where each word is modeled as draw different multinomial distribution depending on the topic assignment of that word. The representation is often used together with preprocessing such as stemming and stop words. An Illustration of how text is preprocessed and represented in the bag of words model is given in Figure 2.1.

2.2 Computational Bayesian inference

This section begins with an introduction to Bayesian inference and then comes an explanation of Gibbs sampling as a way of doing computational inference.

Text 1: A Look into the thriving dark web criminal market
Text 2: North Korean hackers tied to cryptocurrency attacks in South Korea

lowercase, stemming, stop words

Vocabulary

	attack	crimin	cryptocurr	dark	hacker	korea	look	market	north	south	thrive	tie	web
Text 1:	0	1	0	1	1	0	1	1	0	0	1	0	1
Text 2:	1	0	1	0	0	2	0	0	1	1	0	1	0

Figure 2.1: Illustration of how text is preprocessed and represented in the bag of words model. The red words are stop words that are removed in the preprocessing step.

2.2.1 Bayesian inference

We have observed data y that is distributed as $F(\theta)$, a distribution parameterized by θ. The goal in Bayesian inference is to calculate the posterior distribution $p(\theta|y)$ of the parameter θ given the observed data. Using Bayes rule to expand we have

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)} \propto p(\theta)p(y|\theta), \tag{2.1}$$

in the last step we used that $p(y)$ is not dependant on θ and is only a normalizing constant. We see that the posterior distribution is the product of the prior $p(\theta)$ and the likelihood $p(y|\theta)$. The prior is our prior belief of the parameters and is specified as part of the model. The likelihood $p(y|\theta)$ is the probability of observing the data given θ.

2.2.2 Gibbs sampling

It is often intractable to calculate and sample from the exact posterior distribution, so instead simulation is used to obtain samples from the posterior. Markov Chain Monte Carlo is a computation method for drawing values from arbitrary posterior distribution. The goal is to construct as sequence of samples $\theta^{(1)}, \theta^{(2)}, \ldots$ where each value only depends on the previous one, so that the sequence forms a Markov chain. The construction is done so that the Markov chain converges to the desired posterior distribution. Gibbs sampling is an example of a Markov Chain Monte Carlo algorithm.

In Gibbs sampling the parameter vector $\theta = (\theta_1, \theta_2, \ldots, \theta_d)$ is divided into d subcomponents. In each iteration t each of the components is updated by sampling from the conditional distribution given the values of all other components

$$p(\theta_i^{(t)}|\theta_1^{(t)}, \ldots, \theta_{i-1}^{(t)}, \theta_{i+1}^{(t-1)}, \ldots, \theta_d^{(t-1)}), \tag{2.2}$$

this is repeated for $t = 1, 2, \ldots$ until it has converged. The Gibbs sampling algorithm is described in Algorithm 2.1. Assessing convergence can be difficult and for many application Gibbs sampling is just done for a fixed number of iterations and is then assumed to have converged.

Algorithm 2.1 General Gibbs sampling algorithm

```
Initialize θ_i^(0) randomly
t ← 1
while not converged do
    for i = 1, ..., d do
        Sample θ_i^(t) according to Equation (2.2)
    end
    t ← t + 1
end
```

2.3 Dirichlet process

2.3.1 Dirichlet distribution

The Dirichlet distribution is the conjugate prior distribution of the multinomial distribution. Therefore, draws from the Dirichlet distribution is a vector $\theta = (\theta_1, \ldots, \theta_K)$ with constraint $\sum_{i=1}^{K} \theta_i = 1$. The Dirichlet distribution is parameterized by parameters $\alpha_1, \ldots, \alpha_K$ and has probability density function

$$p(\theta) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1} \propto \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}. \tag{2.3}$$

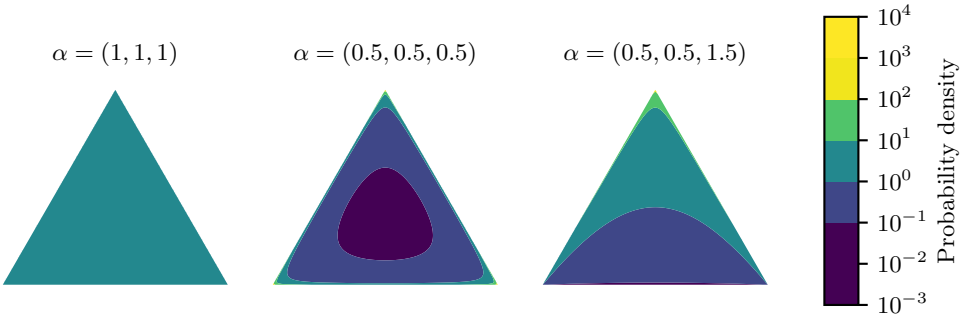


Figure 2.2: Example of probability density function for different parameters for the Dirichlet distribution. The triangle represents the surface of a 2-simplex and the distribution is in three dimensions. In the figure it can be seen how parameters $\alpha_i < 1$ favors sparse vectors and that $\alpha_i = 1$ gives same probability for all vectors.

Letting $y = (y_1, \ldots, y_n) \sim \text{Multinomial}(\theta_1, \ldots, \theta_K)$ we get the posterior

$$p(\theta|y) \propto p(y|\theta)p(\theta) \propto \prod_{i=1}^{K} \theta_i^{\alpha_i + y_i - 1}, \tag{2.4}$$

the prior parameters α_i can be seen as pseudo observations. A uniform prior density is achieved by setting $\alpha_i = 1$ for all i, which gives all possible vectors the same density. For $\alpha_i < 1$ we can not have the same interpretation, but these lower values will assign higher density to vectors where the probability is concentrated to a few categories. How the parameters effect the probability density function is illustrated for the 3-dimensional case in Figure 2.2.

2.3.2 Dirichlet process

The Dirichlet process (DP) is a infinite extension of the Dirichlet distribution. The Dirichlet process is a distribution over distributions. Therefore a distribution G can be a draw from a DP,

$$G \sim \text{DP}(\gamma, H), \tag{2.5}$$

here the DP is parameterized by the concentration parameter γ and base distribution H. The distribution G is discrete even if the base distribution H is continuous. This fact and the effect of γ is illustrated in Figure 2.3 where example of realisations is show for different γ.

Drawing i independent identically distributed variables from G we have $\theta_j|G \sim G$ for $j = 1, \ldots, i$. It can be shown [6, 4] that we for θ_i have the following conditional probability

$$\theta_i|\theta_1, \ldots, \theta_{i-1} \sim \frac{1}{i - 1 + \gamma} \sum_{j=1}^{i-1} \delta_{\theta_j} + \frac{\gamma}{i - 1 + \gamma} H. \tag{2.6}$$

Due to the exchangeability for the draws θ_i [6] and using the notation $\theta_{-i} = \{\theta_j, j \neq i\}$ we can get that

$$\theta_i|\theta_{-i} \sim \sum_{j=1}^{k} \frac{n_j}{i - 1 + \gamma} \delta_{\theta_j} + \frac{\gamma}{i - 1 + \gamma} H, \tag{2.7}$$

where n_j is the number times θ_j has been observed, and k is the number of different θ_j. The larger n_j (more observations in a cluster) the larger probability for observing that cluster. This rich-gets-richer phenomenon, makes the DP a suitable prior for clustering problems.

Since each draw is independent we can use the linearity of expectation to calculate expectation for the number of clusters m for a DP with n draws

$$\text{E}(m) = \sum_{i=1}^{n} \text{E}(\text{new cluster draw } i) = \sum_{i=1}^{n} \frac{\gamma}{i - 1 + \gamma} \leq \sum_{i=1}^{n} \frac{\gamma}{n} \leq \mathcal{O}(\gamma \log n) \tag{2.8}$$

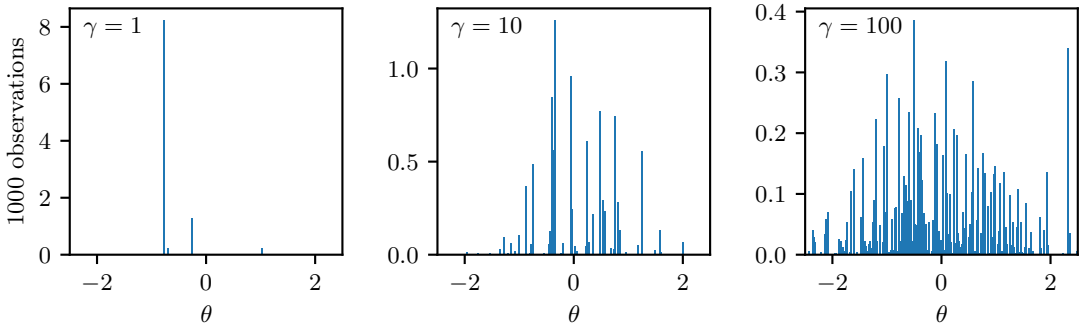


Figure 2.3: Histogram of 10 000 draws $\theta_i \sim G$ where $G \sim \mathrm{DP}(\gamma, \mathcal{N}(0,1)))$ for different values of γ. The draws was generated using the Chinese restaurant process view of the DP.

in the last step we used the fact that the harmonic series is limited by the logarithmic function. We can see that the number of expected clusters is much lower than n and is in fact of the order of $\log n$.

2.3.3 Chinese restaurant process

The Chinese restaurant process (CRP) is a commonly used metaphor and representation of DP explaining the clustering property of the DP.

In the metaphor we have a restaurant with infinite number of tables and each table can seat infinite number of customers. The first customer sits at table with dish θ_1 and the second customer sits at the same table with probability $\frac{1}{\gamma+1}$ and a new table with probability $\frac{\gamma}{1+\gamma}$. The following customers sit down at an already existing table with probability proportional to the number of customer at that table and a new table with probability proportional to γ. The resulting conditional probability for θ_i given the previous $\theta_1, \ldots, \theta_{i-1}$ becomes

$$\theta_i | \theta_1, \ldots, \theta_{i-1} \sim \frac{1}{i-1+\gamma} \sum_{j=1}^{i-1} \delta_{\theta_j} + \frac{\gamma}{i-1+\gamma} H, \tag{2.9}$$

where δ_{θ_j} is the distribution concentrated at θ_j. We see that this is the same conditional probability as in Equation (2.6), and therefore the procedure does generate draws from a DP.

2.4 Dirichlet process mixture models

In this section we introduce how to use DP as a prior in a mixture model and how it corresponds to the infinite limit of a finite Dirichlet mixture model. Then a Gibbs sampling algorithm for inference is introduced.

We have data points $y_1, \ldots, y_n$ that we assume are interchangeable, each data point y_i can be multivariate. We model y_i as being drawn from a mixture of distributions $F(\theta)$, with θ being distributed G. We then assume G to be drawn from a DP with concentration parameter γ and base distribution H, so the model is

$$\begin{aligned} y_i | \theta_i &\sim F(\theta_i) \\ \theta_i | G &\sim G \\ G &\sim \mathrm{DP}(\gamma, H). \end{aligned} \tag{2.10}$$

2.4.1 As an Infinite mixture

An equivalent model to Equation (2.10) can be achieved by taking the limit $K \to \infty$, where K is the number of components of a Dirichlet mixture model. The Dirichlet mixture models the data as being a mixture of K distributions $F(\phi_{c_i})$. The latent variables c_i are drawn from a discrete

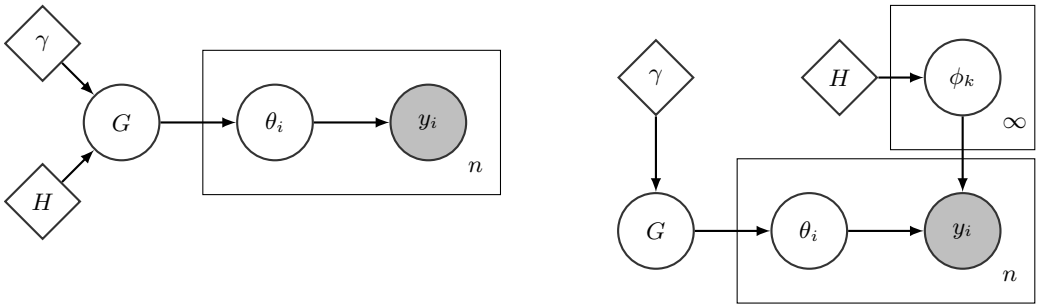


Figure 2.4: Plate diagram showing the Bayesian model for DP left and for the infinite limit of the finite mixture right. Diamonds are hyperparameters, circles are latent variables and filled circles are observed variables.

distribution with a symmetric Dirichlet prior, with concentration parameter γ/K. The parameters ϕ_c draw from a distribution H and the Dirichlet mixture model is

$$\begin{aligned} y_i|c_i, \boldsymbol{\phi} &\sim F(\phi_{c_i}) & c_i|\boldsymbol{p} &\sim \text{Discrete}(\boldsymbol{p}) \\ \boldsymbol{p} &\sim \text{Dirichlet}(\gamma/K, \ldots, \gamma/K) & \phi_c &\sim H. \end{aligned} \tag{2.11}$$

By integrating over the mixing proportions $\boldsymbol{p}$ the conditional probability for class assignment is

$$p(c_i = c|c_{-i}) = \frac{m_c^{(i)} + \gamma/K}{i - 1 + \gamma}, \tag{2.12}$$

where $m_c^{(i)}$ is the number of $c_j = c$ for all $j < i$. Taking the limit $K \to \infty$ we get

$$p(c_i = c|c_{-i}) \xrightarrow[K\to\infty]{} \begin{cases} \dfrac{m_c^{(i)}}{i - 1 + \gamma} & \text{if } c_j = c \text{ for one } j = i \\ \dfrac{\gamma}{i - 1 + \gamma} & \text{the rest} \end{cases}, \tag{2.13}$$

With the conditional probabilities implied by this limit and denoting $\phi_{c_i} = \theta_i$ we can see that this model in the limit is equivalent to the Dirichlet process model in Equation (2.10) [4]. The construction of the DPM as the limit of the Dirichlet model is helpful for understanding the model and when deriving Gibbs sampling algorithm for the model. The difference between the infinite mixture and the DP model represented as a graphical models can be seen in Figure 2.4.

2.4.2 Gibbs sampling algorithm

For a model there is more than one possible Gibbs sampling algorithm, which may have different convergence rates. To derive a Gibbs sampling algorithm we use Bayes rule for the following conditional probability

$$\begin{aligned} p(c_i = c|c_{-i}, y_i, \boldsymbol{\phi}) &\propto p(c_i = c|c_{-i}, \boldsymbol{\phi}) p(y_i|c, \boldsymbol{\phi}) \\ &\propto p(c_i = c|c_{-i}) p(y_i|\phi_c). \end{aligned} \tag{2.14}$$

In the second step we used for the first term the fact that c_i is independent of $\boldsymbol{\phi}$ and for second that y_i is given c_i only dependant of ϕ_{c_i}. In the limit $K \to \infty$ we only represent those ϕ_c with some observation and we get

$$p(c_i = c|c_{-i}, y_i, \boldsymbol{\phi}) \propto \begin{cases} \dfrac{m_c^{(i)}}{i - 1 + \gamma} p(y_i|\phi_{c_i}) & \text{if } c_j = c \text{ for some } j \neq i \\ \dfrac{\gamma}{i - 1 + \gamma} \displaystyle\int p(y_i|\phi) \, \mathrm{d}H(\phi) & c \text{ new cluster} \end{cases}, \tag{2.15}$$

So for the classes with no previously associated data points we have that the likelihood is the predictive prior likelihood. The predictive prior likelihood is evaluated as

$$\int p(y_i|\phi)\,\mathrm{d}H(\phi) = \int p(y_i|\phi)p(\phi)\,\mathrm{d}\phi, \tag{2.16}$$

if it is analytically solvable it can be used to define a Gibbs sampling algorithm. The Gibbs sampler would have the state is represented by $c_1, \ldots, c_i$ and $\{\phi_c : c \in c_1, \ldots, c_i\}$. Sampling c_j according to Equation (2.15) and each ϕ_c from the posterior distribution based on the prior H the observations y_j for which $c_i = c$. For more details about this and other possible Gibbs sampling schemes for DPMM see [4].

It is also possible to integrate out ϕ_c and eliminate them from the algorithm. By doing this we instead have

$$p(c_i = c|c_{-i}, y_i) \propto \begin{cases} \dfrac{m_c^{(i)}}{i-1+\gamma} \displaystyle\int p(y_i|\phi)\,\mathrm{d}H_c^{(i)}(\phi) & \text{if } c_j = c \text{ for some } j \neq i \\ \dfrac{\gamma}{i-1+\gamma} \displaystyle\int p(y_i|\phi)\,\mathrm{d}H(\phi) & c \text{ new cluster} \end{cases}, \tag{2.17}$$

here $H_c^{(i)}$ is the posterior for ϕ_c given prior H and the observations y_j for which $c_i = c$. Here the state is the cluster assignments $c_1, \ldots, c_n$ and the algorithm is that each iteration draw c_i for $i = 1, \ldots, n$ from Equation (2.17). For this Gibbs sampling algorithm to be feasible both the integrals in Equation (2.17) needs to be analytically computable. This is the case when the distribution H is conjugate prior with F. There are other possible sampling schemes for DPM when dealing with non-conjugate priors. This is not explored further in this thesis as conjugate priors will be used. For more information about handling non-conjugate see [4].

2.4.3 DP multinomial mixture (DPMM) for topic modeling

We have a set of interchangeable documents $y_1, \ldots, y_N$ represented by their word vectors as described in the bag of words model. Each document is assumed to be a sample drawn from a multinomial distribution. To have a conjugate prior we choose a Dirichlet distribution with symmetric concentration parameters α for the base measure H. The model is described as

$$\begin{aligned} y_i|\theta_i &\sim \text{Multinomial}(\theta_i) \\ \theta_i|G &\sim G \\ G &\sim \text{DP}(\gamma, \text{Dirichlet}(\alpha)) \end{aligned}. \tag{2.18}$$

To use the Gibbs sampler in Equation (2.17) for this model we need to calculate the posterior H_c^i and calculate the two integrals. The posterior for ϕ_c is easily calculated for conjugate prior and we have

$$\phi_c|y_j \text{ where } j \neq i \text{ and } c_j = c \sim \text{Dirichlet}(\alpha + n_{c,1}^{(i)}, \ldots, \alpha + n_{c,V}^{(i)}), \tag{2.19}$$

where $n_{-i,c,j}$ is the number of word j is observed in topic c without observation i. The predictive likelihood becomes a Dirichlet-Multinomial distribution and the integral is calculated as

$$\int p(y_i|\phi)\,\mathrm{d}H_c^{(i)}(\phi) = \frac{n!}{\prod_{k=1}^V y_k!} \frac{\prod_{k=1}^V \prod_{j=1}^{y_k} (j - 1 + \alpha + n_{-i,c,k})}{\prod_{k=1}^n (k - 1 + \alpha V + n_{-i,c})}, \tag{2.20}$$

where $n_{-i,c}$ is the total number of words in topic c without observation i. For exact details for computation of the integral see Appendix A. For a new cluster without any observation the integral is also a Multinomial likelihood integrated over a Dirichlet prior, so we get

$$\int p(y_i|\phi)\,\mathrm{d}H(\phi) = \frac{n!}{\prod_{k=1}^V y_k!} \frac{\prod_{k=1}^V \prod_{k=1}^{y_k} (k - 1 + \alpha)}{\prod_{k=1}^n (k - 1 + \alpha V)}. \tag{2.21}$$

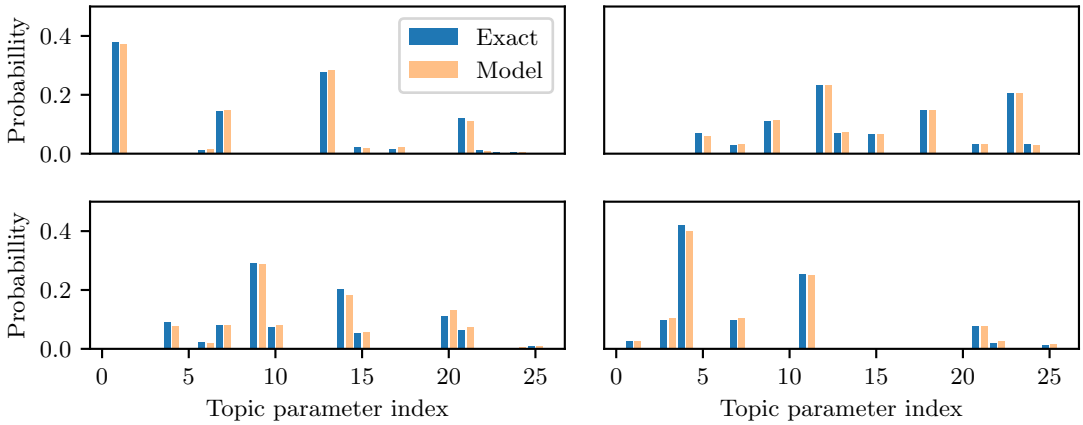


Figure 2.5: Each of the panels shows the exact topic parameters compared with the topic parameters discovered by the inference by DPMM model. It can been seen that the model successfully recovered the topic parameters and number of topics.

Combining this with Equation (2.17) we have the conditional probability

$$p(c_i = c|c_{-i}, y_i) \propto \begin{cases} \dfrac{m_c^{(i)}}{i-1+\gamma} \dfrac{\prod_{k=1}^{V} \prod_{j=1}^{y_k} (j-1+\alpha+n_{c,k}^{(i)})}{\prod_{k=1}^{n} (k-1+\alpha V+n_c^{(i)})} & \text{if } c_j = c \text{ for some } j \neq i \\ \dfrac{\gamma}{i-1+\gamma} \dfrac{\prod_{k=1}^{V} \prod_{k=1}^{y_k} (k-1+\alpha)}{\prod_{k=1}^{n} (k-1+\alpha V)} & \text{a new cluster } c \end{cases} . \quad (2.22)$$

The Gibbs sampling algorithm is just to repeatedly sample each c_i using the conditional probability. Before and after each sample the we update the appropriate cluster parameters m_c, $n_{c,k}$, n_c.

2.4.4 Example using DPMM

To illustrate that the DPMM Gibbs sampling algorithm converges works it was tested using generated data. The data was generated by first drawing 4 topic parameters $\phi_i \sim \text{Dirichlet}(0.1)$ in a vocabulary of 100 words, then 1000 observations y_i of length 5 were generated by choosing a random topic θ_i with equal probability and drawing $y_i \sim \text{Multinomial}(\phi_{\theta_i})$.

Gibbs sampling was done using 1000 iterations for a DPMM with hyperparameters $\gamma = 1.0$ and $\alpha = 0.1$. The result was the model discovered 4 topics that are compared with the exact topics in Figure 2.5. It is observed that the model successfully recovered both the number of topics and their distributions.

2.5 Evaluation of topic models

2.5.1 Perplexity

When the documents in the corpus are unlabeled, topic modeling can be seen as doing density estimation. A common way of evaluating topic models is to compare likelihood on a held-out test set. For topic models the likelihood is commonly compared using perplexity [7, 8]. Perplexity is monotonically decreasing in the likelihood of the test set. Lower perplexity indicates better performance since it corresponds to higher likelihood. For a test set D_{test} consisting of M documents the perplexity is

$$\text{perplexity}(D_{\text{test}}) = \exp\left(-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\right), \quad (2.23)$$

where N_d is the length of each document and w_d is the word vector for each document.

In [9] it is shown that perplexity does not always reflect how humans measure the coherence of a topic. Therefore other topic coherence measures have been proposed in [10] to address this. The measures utilize Wikipedia as a reference corpus to capture the relation of words. But the measures do not solve all problems as they only evaluate on the top words of each topic and not the full model. They can not be used to evaluate non-word data such as hashtags if they are not used in the reference corpus.[1]

[1]The tool Palmetto [11], which was released with the paper [10], does not reproduce the numbers in the paper, see `https://github.com/dice-group/Palmetto/issues/13`. Also the best coherence measure from the paper seems to have undesirable properties, see `https://github.com/dice-group/Palmetto/issues/12`. So it is unclear if the measures were fully evaluated in [10].

Chapter 3

Evolutionary Dirichlet process

In this chapter we first introduce the Evolving Dirichlet process (EDP) and extension of DPs to handle temporal data. Then we introduce a inference algorithm based on Gibbs sampling for EDP and how it can be applied to topics modeling. The way of extending DPs to handle temporal data and modeling of topic parameters is our contribution to the field of topic modeling. The way it is done is inspired by the extension described in [8]. By having a mixture in the base measure instead of a mixture of DPs as in [8] we create a more stable evolutionary model.

3.1 Problem setting

We have a ordered dataset $Y = \{Y_1, \ldots, Y_T\}$ where T is the number of epochs. The set of M_t observations in epoch t is $Y_t = \{y_{t,1}, \ldots, y_{t,M_t}\}$. The observations within an epoch are assumed to be interchangeable while observations between epochs are not. The goal is to cluster observations within an epoch and find traces that shows how clusters in the current epoch are dependant on the ones in the previous epoch. Clusters can be born in an epoch, split into several clusters in the next epoch or die out.

3.2 Evolutionary Dirichlet process

Each observation $y_{t,i}$ is modeled as being drawn from a mixture of distributions $F(\theta_{t,i})$, with $\theta_{t,i}$ being distributed G. We then assume G to be drawn from a DP with concentration parameter γ and base measure that is a mixture of distribution H_0 and the distributions $H_{t-1,p}$ based on the clusters in the previous epoch. The weight of the base distribution H_0 is chosen to be a hyperparameter μ and the weight of the posteriors proportional to the number of observations assigned to the cluster in the previous epoch. Therefore the model for each epoch is

$$\begin{aligned} y_{t,i}|\theta_{t,i} &\sim F(\theta_{t,i}) \\ \theta_{t,i}|G &\sim G \\ G &\sim \mathrm{DP}\left(\gamma, \mu H_0 + \frac{1-\mu}{M_{t-1}} \sum_{p=1}^{N_{t-1}} m_{t-1,p} H_{t-1,p}\right). \end{aligned} \tag{3.1}$$

This model for each epoch is the same as the static DP mixture model except for that the base measure is a mixture of terms. The base measure is influenced by clusters in previous epochs, allowing us to recover relationships between topics in the current epoch and those in previous epochs. These relationships are defined as follows:

- A cluster is born in epoch t if it has prior H_0.
- A cluster in epoch t inherits from topic p in epoch $t-1$ if it has prior $H_{t-1,p}$.
- A clusters splits in epoch t if multiple topics inherits from the same topic p in epoch $t-1$.
- A cluster dies in epoch t if no clusters inherits from it in epoch $t+1$.

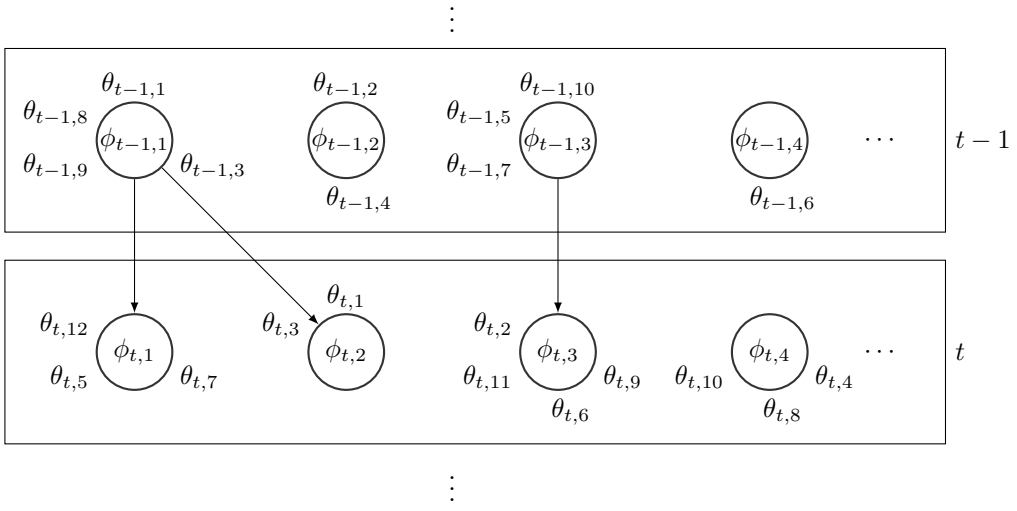


Figure 3.1: Chinese restaurant process representation of the EDP. The tables are represented by circles and dish i at epoch t by $\phi_{t,i}$. Customer i at day t is represented by $\theta_{t,i}$ and is sitting next to the assigned table. Arrows between epochs represent that a dish is influenced by the dish in the previous epoch.

3.2.1 Construction using Chinese restaurant process

Construction of EDP can be done using Chinese restaurant processes. In this metaphor each epoch is a fixed period, say a day, in a Chinese resultant with infinite tables and each table can seat an infinite amount of customers. All customers only stay at a restaurant during one day. At the end of a day the owner analyses the dishes served and their consumption and based on that information changes the special dish menu to include variants of those dishes.

The generative process is the following. Customer $\theta_{t,i}$ enters, sits at one of the existing tables in restaurant t with probability proportional to the number of customer at that table $m_{t,p}$ or he sits at a new table with a probability of $\frac{\gamma}{i-1+\gamma}$. If the customer sits at en existing table he shares of the dish at that table. If he sits at a new table he chooses a new dish from the base menu H_0 with probability μ or a variation of a dish $H_{t-1,p}$ from the previous day with probability proportional to the number of customers sharing that dish the previous day $m_{t-1,p}$. Putting it together the conditional probability can be written as

$$\theta_i|\theta_1,\ldots,\theta_{i-1} \sim \frac{1}{i-1+\gamma}\left(\sum_{p=1}^{N_t} m_{t,p}\delta_{\phi_{t,p}} + \gamma\left(\mu H_0 + \frac{1-\mu}{M_{t-1}}\sum_{p=1}^{N_{t-1}} m_{t-1,p}H_{t-1,p}\right)\right), \tag{3.2}$$

which we will see is the same conditional probability as achieved by the DP model. The Chinese restaurant process gives an equivalent more easily interpretable description of the model.

3.3 Gibbs sampling for EDP

In this section we derive a collapsed Gibbs sampling algorithm for the EDP model. The Gibbs sampler is derived by considering the infinite limit of the equivalent finite mixture model. For the algorithm we need to derive the conditional probability for the cluster assignments and the conditional probability for the prior assignments.

3.3.1 The infinite limit of a finite mixture model

In order to derive a Gibbs sampling scheme for the model we consider the equivalent finite mixture model in the limit of the number of clusters $K \to \infty$. The equivalent model is described as

$$\begin{aligned} y_{t,i}|\theta_{t,i}, \boldsymbol{\phi} &\sim F(\phi_{\theta_{t,i}}) \\ \theta_{t,i}|\boldsymbol{p} &\sim \text{Categorical}(\boldsymbol{p}) \\ \phi_{t,j}|z_{t,j} &\sim H_{t-1,z_{t,j}} \\ z_{t,j} &\sim \text{Categorical}\left(\mu, (1-\mu)\frac{m_{t-1,1}}{M_{t-1}}, \ldots, (1-\mu)\frac{m_{t-1,N_{t-1}}}{M_{t-1}}\right) \\ \boldsymbol{p} &\sim \text{Dirichlet}(\gamma/K, \ldots, \gamma/K) \end{aligned}, \tag{3.3}$$

we have the extra latent variables $z_{t,j}$ indicating the priors for each clusters. All other notation is the same as in the same as in Section 3.2 and is also explained in Table 3.1.

3.3.2 Sampling $\theta_{t,i}$

Sampling $\theta_{t,i}$ is almost the same as for the static DP model from Section 2.4 except for sampling a new cluster. For the new cluster the integration over the mixture of priors can be split into one term for each prior. Each prior will also be weighted with its weight in the mixture. Therefore the full conditional probability is

$$p(\theta_{t,i} = k|\theta_t^{(i)}, z_t, Y_t) \propto \begin{cases} m_{t,k}^{(i)} \int p(y_{t,i}|\phi)\mathrm{d}H_{t,k}^{(i)}(\phi) & \text{existing cluster } k \text{ if } \theta_{t,q} = k \text{ for some } q \neq i \\ \gamma(1-\mu)\frac{m_{t-1,p}}{M_{t-1}} \int p(y_{t,i}|\phi)\mathrm{d}H_{t-1,p}(\phi) & \text{new cluster } k \text{ with prior } H_{t-1,p} \\ \gamma\mu \int p(y_{t,i}|\phi)\mathrm{d}H(\phi) & \text{new cluster } k \text{ with prior } H_0 \end{cases} \tag{3.4}$$

3.3.3 Sampling $z_{t,j}$

For each existing cluster j we need to sample the variable $z_{t,j}$. Considering Bayes rule for the conditional probability for $z_{t,i}$ we have

$$\begin{aligned} p(z_{t,j} = p|z_t^{(j)}, Y_t, \boldsymbol{\phi}) &\propto p(z_{t,j} = p|z_t^{(j)}, \boldsymbol{\phi}) p(\text{all } y_{t,i} \text{ where } \theta_{t,i} = j \mid c, \boldsymbol{\phi}) \\ &\propto p(z_{t,j} = p) p(\text{all } y_{t,i} \text{ where } \theta_{t,i} = j \mid c, \boldsymbol{\phi}), \\ &\propto p(z_{t,j} = p) \prod_{i=1:\theta_{t,i}=j}^{M_t} p(y_{t,i}|\phi_{t,j}). \end{aligned} \tag{3.5}$$

Here $\boldsymbol{\phi}$ is the set of all cluster parameters $\phi_{t,1}, \phi_{t,2}, \ldots$ in the current epoch. In the second step we used that $z_{t,i}$ is independent of $\boldsymbol{\phi}$ and $z_t^{(i)}$ and in the last step we used that all observations $y_{t,i}$ are independent. Integrating out ϕ to get the collapsed conditional probability,

$$p(z_{t,j} = p|z_t^{(j)}, \theta_t, Y_t) \propto \begin{cases} (1-\mu)\frac{m_{t-1,p}}{M_{t-1}} \int \prod_{i=1:\theta_{t,i}=j}^{M_t} p(y_{t,i}|\phi)\mathrm{d}H_{t-1,p}(\phi) & \text{cluster } j \text{ has prior } H_{t-1,p} \\ \mu \int \prod_{i=1:\theta_{t,i}=j}^{M_t} p(y_{t,i}|\phi)\mathrm{d}H(\phi) & \text{cluster } j \text{ is new topic} \end{cases} \tag{3.6}$$

Table 3.1: Symbols used in model.

Symbol	Meaning
V	Number of words of vocabulary
M_t	Number of observations in epoch t
N_t	Number of clusters in epoch t
$m_{t,p}$	Number of observations assigned to cluster p in epoch t
$n_{t,p}$	Total number of words assigned to cluster p in epoch t
$n_{t,p,w}$	Number of times word w has been assigned to cluster p in epoch t
$y_{t,i}$	Observation i in epoch t
$\theta_{t,i}$	Clusters assigned to observation $y_{t,i}$
θ_t	The set of cluster assignments for observations Y_t
$z_{t,j}$	Prior assigned to cluster j
z_t	The set of prior assignments in epoch t
$\phi_{t,j}$	Clusters parameter for topic j in epoch t
$\alpha_{t,p,w}$	Parameter for word w for DP with base distribution $H_{t,p}$
$\alpha_{t,p}$	Sum over prior parameters $\sum_{i=1}^{V} \alpha_{t,p,i}$
$\cdot_{t,\cdot}^{(i)}$	Other variable without considering observation $y_{t,i}$

3.4 Topic modeling

In this section we introduce the EDP multinomial mixture (EDPMM) where EDP is used for topic modeling. We explain how the priors are affected by the last epoch clusters and then the integrals in the Gibbs algorithm probabilities are calculated.

3.4.1 Modeling topic parameters

When using EHP for topics modeling we represent documents using the bag of words model and realisations from a multinomial distribution. So the distribution F in Equation (3.1) is a multinomial distribution. Conjugate prior is used so the base measure H_0 is a symmetric Dirichlet distribution with parameter Dirichlet(α). Left to specify is how the clusters in the previous epoch affect the prior in the current epoch. The base measures $H_{t-1,p}$ are chosen to be Dirichlet distributions (with dimension V, the size of the vocabulary.)

$$H_{t-1,p} = \text{Dirichlet}(\alpha_{t-1,p,1}, \ldots, \alpha_{t-1,p,V}), \tag{3.7}$$

so they are Conjugate priors. The parameters $\alpha_{t-1,p,w}$ are chosen to be the posterior of cluster p times a decay factor λ for the observations

$$\alpha_{t-1,p,w} = \alpha + \lambda(\alpha_{t-2,p_{t-1},w} - \alpha + n_{t-1,p,w}), \tag{3.8}$$

where $n_{t-1,p,w}$ is the number of times word w was assigned to topic p in epoch $t-1$ and $\alpha_{t-2,p_{t-2},w}$ is the prior for topic p in epoch $t-1$. So for a topic born in t_b with parents with indices $p_{t_b}, \ldots, p_{t-1}$ in each of the epochs $t_b, \ldots, t_{t-1}$ we have

$$\alpha_{t-1,p,w} = \alpha + \sum_{i=t_b}^{t-1} \lambda^{t-i} n_{i,p_i,w}, \tag{3.9}$$

this is Equation (3.8) expanded until the topic birth. We see that the prior is dependant on all prior clusters weighted on how far they are from the current epoch.

3.4.2 Gibbs sampling EDPMM

We conjugate Dirichlet priors for the topics parameters and therefore the integrals in Equation (3.4) are traceable as shown in Appendix A. Performing the integrals in Equation (3.4) we get

$$p(\theta_{t,i} = k|\theta_{t-1}, \theta_t^{(i)}, z_t, Y_t) \propto$$
$$\begin{cases} m_{t,k}^{(i)} \dfrac{\prod_{w \in y_{t,i}} \prod_{j=1}^{N_d^w} (n_{t,k,w}^{(i)} + \alpha_{t-1,z_{t,q},w} + j - 1)}{\prod_{i=1}^{N_d} (n_{t,k}^{(i)} + \alpha_{t-1,z_{t,p}} + i - 1)} & \text{if } \theta_{t,q} = q \text{ for some } q \neq i \\ \gamma(1-\mu) \dfrac{m_{t-1,p}}{M_{t-1}} \dfrac{\prod_{w \in y_{t,i}} \prod_{j=1}^{N_d^w} (\alpha_{t-1,p,w} + j - 1)}{\prod_{i=1}^{N_d} (\alpha_{t-1,p} + i - 1)} & \text{new cluster } k \text{ with prior } H_{t-1,p} \\ \gamma\mu \dfrac{\prod_{w \in y_{t,i}} \prod_{j=1}^{N_d^w} (\alpha + j - 1)}{\prod_{i=1}^{N_d} (V\alpha + i - 1)} & \text{new cluster } k \text{ with prior } H_0 \end{cases}, \tag{3.10}$$

where $w \in y_{t,i}$ means for all non zero values in the vector $y_{t,i}$ since only non-zero terms contribute to the likelihood. This is to emphasize that the calculation of probability scales with the number of words in observations, not with the size of the vocabulary.

The conditional probability for the priors indicator $z_{t,j}$ integrate over many multinomial observations. The likelihood of many multinomial observations can be seen as single observation with all the observations combined. Therefore, we have conjugate priors and the integrals are solvable. Performing the integrals in Equation (3.6) we get

$$p(z_{t,j} = p|z_t^{(j)}, \theta_t, Y_t) \propto$$
$$\begin{cases} (1-\mu) \dfrac{m_{t-1,p}}{M_{t-1}} \dfrac{\prod_{w=1}^{V} \prod_{j=1}^{n_{t,j,w}} (\alpha_{t-1,p,w} + j - 1)}{\prod_{i=1}^{n_{t,p}} (\alpha_{t-1,p} + i - 1)} & \text{cluster } j \text{ has prior } H_{t-1,p} \\ \mu \dfrac{\prod_{w=1}^{V} \prod_{j=1}^{n_{t,j,w}} (\alpha + j - 1)}{\prod_{i=1}^{n_{t,i}} (V\alpha + i - 1)} & \text{cluster } j \text{ is new topic} \end{cases}. \tag{3.11}$$

Gibbs sampling involves repeatedly sampling cluster assignments for all observations and sampling priors for currently active topics. For each observation sample we update the bookkeeping variables $n_{t,i}$ and $n_{t,i,w}$. Between epochs we need to calculate the priors for the next epoch and keep track of which topics inherits from previous topics. The Gibbs sampling algorithm for EDPMM can been seen in Algorithm 3.1.

Algorithm 3.1 Gibbs sampling algorithm for EDPMM

Input: Corpus of documents, hyperparameters $\alpha, \gamma, \mu, \lambda$
Output: Topic assignments, topic distributions and topic tree evolution structure

```
for t = 1, ..., T do
    Initialize θ_t randomly
    for j = 1, ..., N_iter do
        for i = 1, ..., N_t do
            Draw θ_{t,i} according to Equation (3.10)
            Update n_{t,θ_{t,i}}, n_{t,θ_{t,i},w} ∀w ∈ y_{t,i}
        end
        for i = 1, ..., M_t do
            Draw z_{t,i} according to Equation (3.11)
        end
    end
    Record relation of new topics to previous topics
    Calculate priors for next epoch according to Equation (3.8)
end
```

3.4.3 Interpretation of hyperparameters

Here we discuss the interpretation and effects the hyperparameters γ, α, μ and λ have on the model. First, γ the concentration parameter to the DP relates to the expected number of topics as can been seen in Equation (2.8). In the limit $\gamma \to 0$ we will only have a single topic and the other limit $\gamma \to \infty$ every topic will only have a single observation. Therefore γ will affect the number of topics discovered by the model.

The parameter α changes the symmetric Dirichlet prior for the topic's words distribution. As seen in Figure 2.2 when $\alpha < 1$ the Dirichlet distribution assigns higher probability to vectors with large weight for a few words. For $\alpha > 1$ the prior assigns higher probability to words distributions that assigns weight to many words. But even for $\alpha > 1$ the model will still strive towards sparse vectors as it is inherit to how sampling is done in the model. As seen in Equation (3.10), an observation will be more likely to be assigned to a topic where words from the observation are common, which when the observation is assigned to it those words will be even more common.

The parameter μ will affect the probability of a topic being assigned the base prior distribution. If $\mu = 1$ all topics in an epoch will be assigned the base prior and no topics will inherit from previous topics. For $\mu = 0$ all topics will inherit from topics in the previous topics and there will be no new topics (except for the first epoch where $\mu = 0$ is not well defined). So for μ closer to 1 more there will be more new topics instead of inheritor topics.

The decay factor λ affects how much the observation from parent topics affects the word distribution of inheritor topics. In the extreme $\lambda = 0$ previous data will have no effect and the model will be as having a static DPMM topics model for each epoch. For $\lambda = 1$ all data from previous epochs will contribute equally to topic word distribution. With $\lambda = 1$ topics parameters can still evolve but changes will be smaller as all previous observations contribute equally.

3.5 Complexity analysis

The most time consuming part in the Gibbs sampling algorithm is calculating the conditional probability in Equation (3.10) and Equation (3.11). The total time complexity for the probability calculations for observations is $\mathcal{O}\left(\tilde{l}M_t K\right)$, where $\tilde{l}$ is the average document length and K the expected number of clusters. For the probability calculation for the priors the total complexity is $\mathcal{O}\left(KVM_{t-1}\right)$. In theory the expected number of clusters drawing M_t observations from a DP is $\mathcal{O}\left(\log M_t\right)$ as shown in Equation (2.8). In practice the number of clusters is dependant on which hyperparameters and the observations, but the expected value still gives some indication. Putting it together we have complexity for each time epoch is $\mathcal{O}\left(\tilde{l}M_t \log M_t\right)$ in expectation.

Chapter 4

Results

4.1 Evaluation on arXiv titles

To evaluate the effectiveness of EDPMM on short texts, we carried out experiments on the titles of the papers published on arXiv [12]. The titles and the categories for each document could be downloaded using their open archive initiative API [13]. The dataset contains 1 375 693 titles between the years 1994 to 2018. The titles where preprocessed in the following steps: (a) converting letters to lowercase, (b) removing stop words and non letter characters, (c) words were stemmed using Snowball [14] (d) removing words shorter that 3 letters, (e) removing words that appear in less than 69 titles (0.005 % of titles). After preprocessing the vocabulary consisted of 7296 words and the titles had an average length of 6.1 words. For perplexity calculations the data set was divided into a training set of 80 % of the titles and a test set with the rest.

4.1.1 Convergence of the Gibbs sampling

To analyses the convergence of the Gibbs sampling algorithm, the perplexity was calculated using different numbers of iterations in each epoch. In Figure 4.1 we see that we have a steady improvement in performance using a higher number of iterations. This indicates that for lower number of iterations the model has not reached a steady sate and the samples are not from the proper posterior distribution.

4.1.2 Comparison with static topic models

The evolutionary model was compared with two non-parametric static topic models; DPMM, a static DP model for each epoch and DPMM-all, a DP model for all data in previous epochs. The idea is here that if the static topic models fit the data better than similar relations as those discovered by EDPMM could be estimated using for example, distance metric between topic parameters. The hyperparameter for EDPMM, DPMM and DPMM-all where $\gamma = 1.0$ for the DP concentration parameters and $\alpha = 0.01$ for topic prior. For EDPMM the extra hyperparameters μ, λ were tuned using a grid search and was set to 0.8 and 0.4 respectively. The number of iterations per epoch was 100 for EDPMM and DPMM and 10 for DPMM-all. The number of iterations for DPMM-all was lower so that the algorithms had similar total running times. In Figure 4.2 we see that EDPMM out performed both the static topic models for all epochs. For both EDPMM and DPMM-all during the first 3 epochs the performance form the extra data is seen. The peak in the last epoch (2018) for DPMM is explained by that there are fewer titles[1] compared with previous year. The evolutionary model and DPMM-all utilize data from previous epochs and do not have a decrease in performance.

The number of topics discovered varies between the models and the difference can be seen in Figure 4.3(b). For the evolutionary model the number of topics increases almost linearly with the number of titles. While the to static models exhibit growth similar to the logarithmic growth

[1] Since the titles were downloaded in May 2018

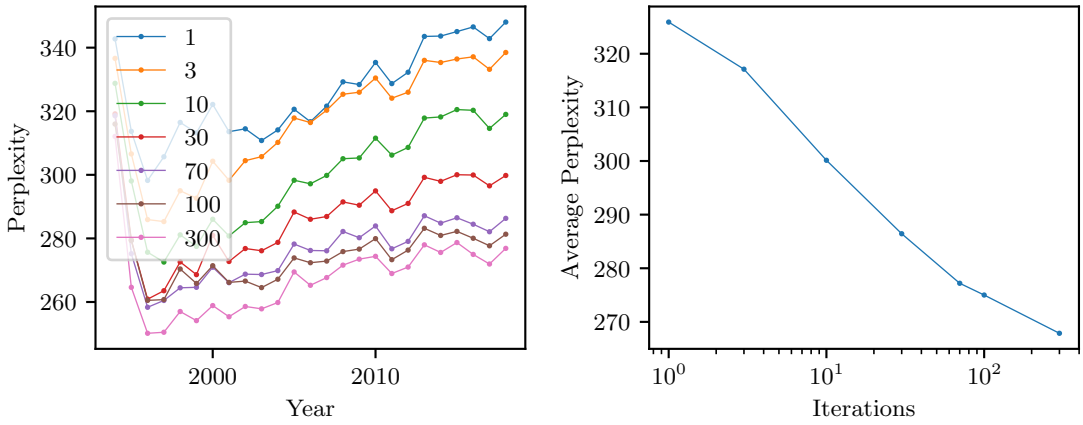


Figure 4.1: Illustration of how perplexity is affected by the number of Gibbs Iterations for EDPMM. Left panel shows perplexity as a function of number of iterations over the epochs and the right panel shows the average perplexity over all epochs as a function of the number of iterations.

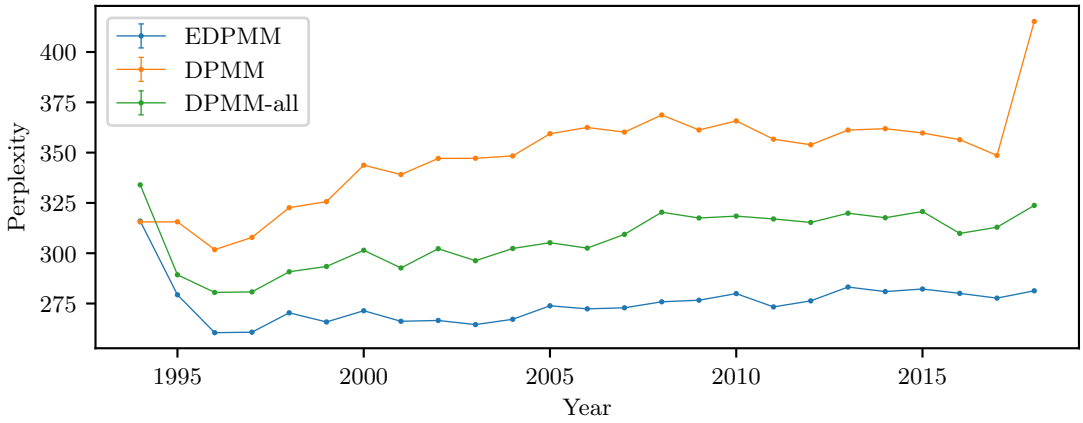


Figure 4.2: Performance of the Evolutionary topic model and static topic models using perplexity. EDPMM is the Evolutionary topic model, DPMM is static topic model for each epoch and DPMM-all is static topic model on the titles for all previous epochs.

expected for DP. How number of titles in the dataset increases over time which can be seen in Figure 4.3(a).

4.1.3 Examples of discovered topics

In Figure 4.4 we can see a topic evolution discovered by the model. The topic evolution shows a single path along the topics tree that ends in a "neural network"[2] topic. The figure also shows the trajectory for some words related to neural networks. The "neural network" topic in Figure 4.4 is only one of many inheritors from the stating topic. So it should not be necessarily considered the direct evolution of the stating topic, rather a evolution of a topic that was a part of the original topic.

In Figure 4.5 we show a partial topic evolutionary tree recovered by the model, the tree is limited to the biggest inheritors in the year 2018. We can see in the figure how the topic is divided into more precise topics that are related to the starting topic.

[2]The name "neural network" was assigned to the topic based on the words in the topic and not something discovered by the model.

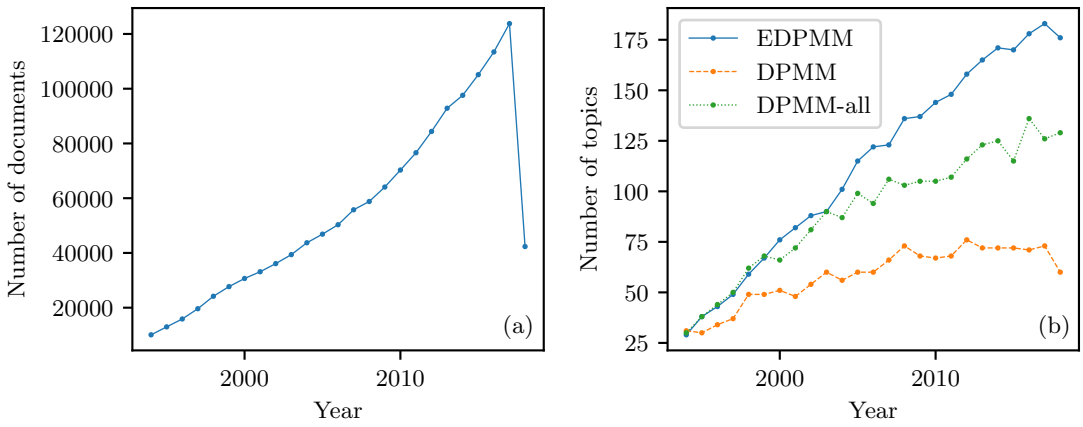


Figure 4.3: Left panel shows how the number of titles grows over time and right panel the number of topics discovered by the models. The evolutionary topics models discovers more topics than both of the topics models.

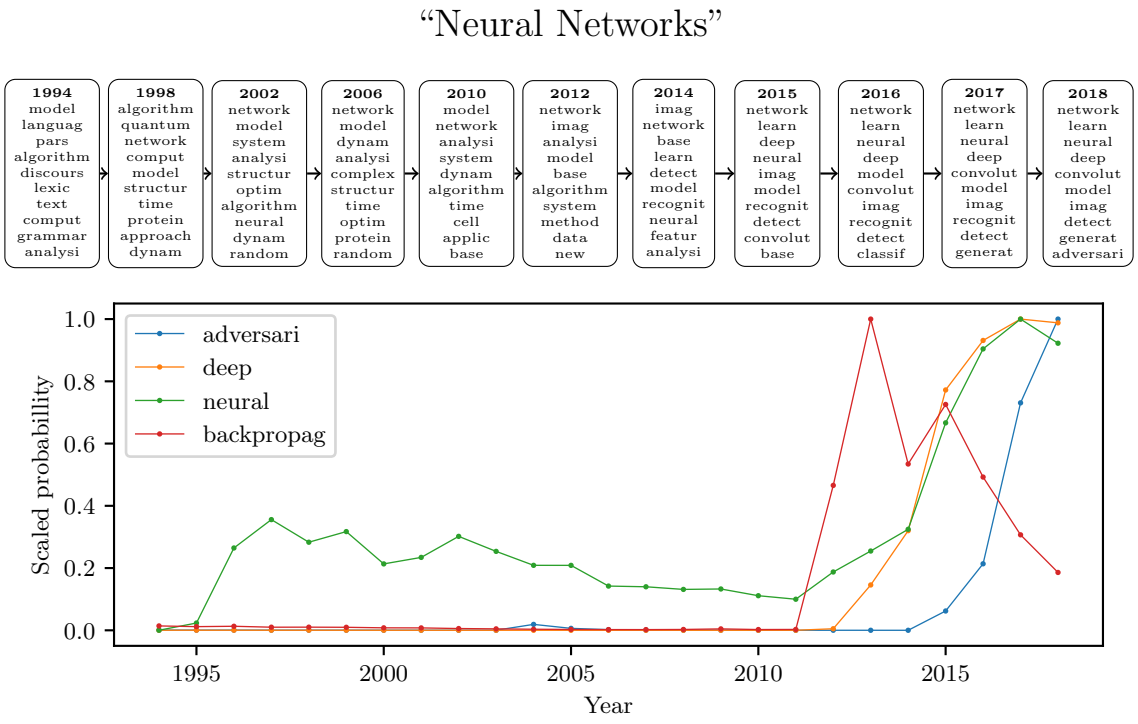


Figure 4.4: Example topic evolution from EDPMM model on arXiv titles dataset. The top panel shows the top ten words from the posterior distribution along the path in the topic tree found by the model. The bottom panel shows the posterior frequency of words as a function of the year, the words probability have been scaled so that the maximum year value for each word is one.

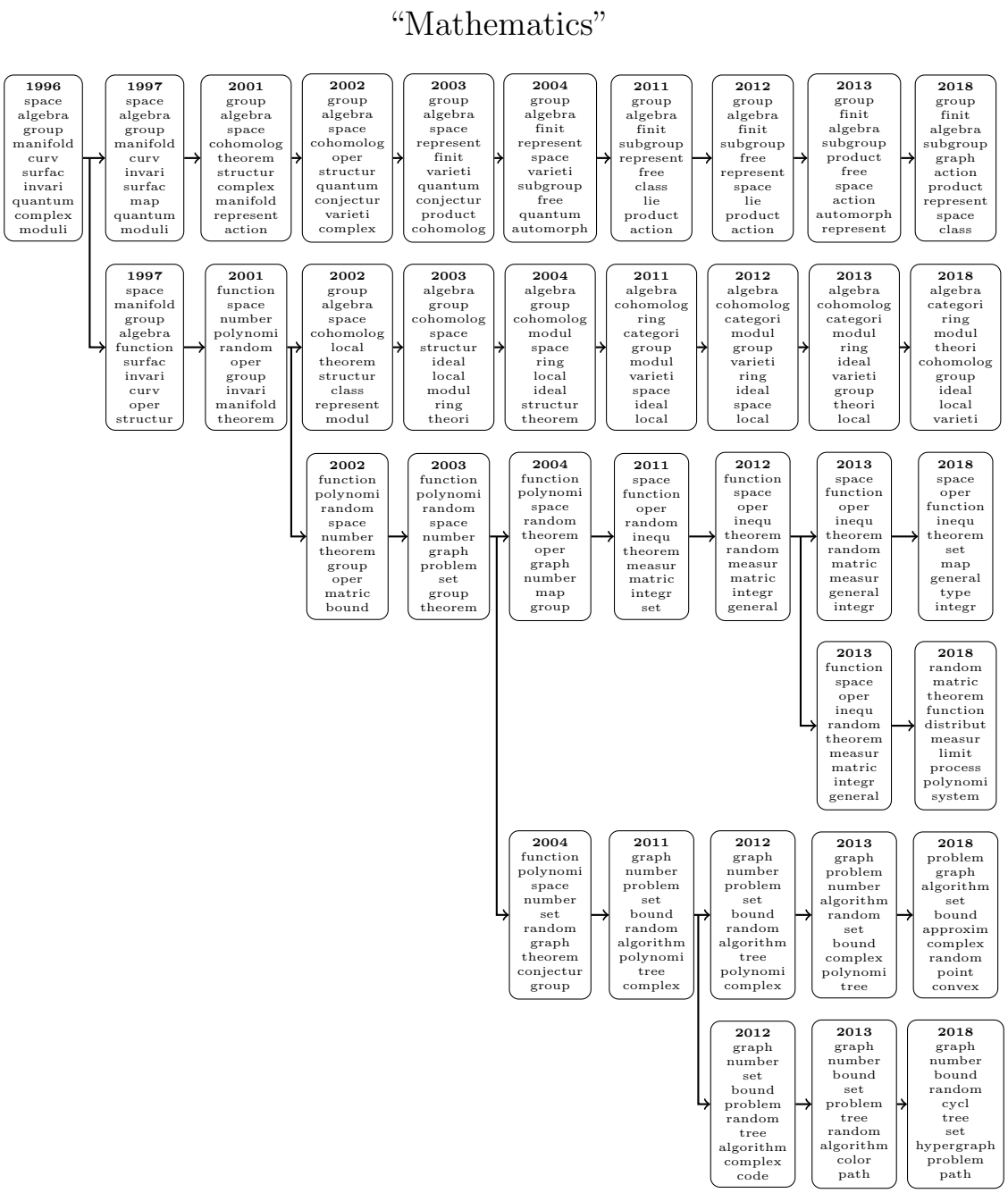
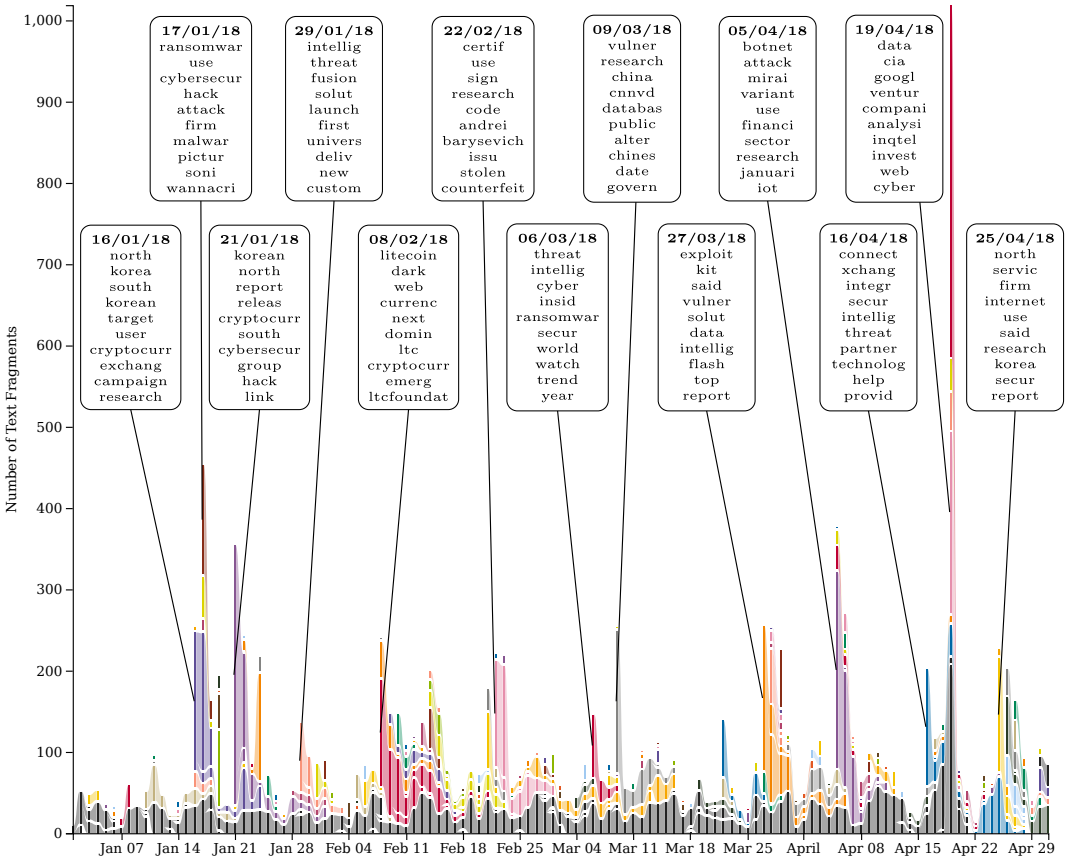


Figure 4.5: Example of topics tree discovered by the EDPMM topic model. The tree shows the six biggest topics in 2018 that inherit from the same topic in 1996. Top words are only shown for years before and after splits due to limited space.

4.2 Evaluation of Recorded Future dataset

To evaluate the effectiveness of EDPMM to discover new topics on short texts, we carried out experiments on text fragments with references to "Recorded Future" extracted from Recorded Future [15]. The text fragments downloaded from Recorded Future include extracts from news sites, social media and many more websites. The dataset contains 11 289 fragments from 01/01/18 to 01/05/18. The fragments where preprocessed in the following steps: (a) converting letters to lowercase, (b) removing stop words and non letter characters, (c) words were stemmed using Snowball [14] (d) removing words shorter that 3 letters, (e) removing words that appear in less than 4 fragments (0.01 % of fragments). After preprocessing the vocabulary consisted of 5423 unique words and the fragments had an average length of 10.9 words. The fragments were divided into epochs per day.

The data was analysed using EDPMM with hyperparameters $\gamma = 1.0$, $\alpha = 0.1$, $\mu = 0.6$ and $\lambda = 0.9$. The parameters μ, λ were tuned using a grid search. In Figure 4.6 shows discovered topic evolution and the top words of some the discovered topics. The tops words are from the posterior on the day that topic first appeared. The figures also list post made on Recorded Future's blog on the same day as the discovered topics. Many of the discovered topics directly correspond to discussion on social media and articles written based on a blog post made on the same day.



Date	Event
16/01/18	North Korea Targeted South Korean Cryptocurrency Users and Exchange in Late 2017 Campaign posted on Recorded Future Blog
29/01/18	Launches Fusion to Deliver the First Universal Threat Intelligence Solution
08/02/18	Litecoin Emerges as the Next Dominant Dark Web Currency posted on Recorded Future Blog
22/02/18	The Use of Counterfeit Code Signing Certificates Is on the Rise posted on Recorded Future Blog
06/03/18	5 Ransomware Trends to Watch in 2018 posted on Recorded Future Blog
09/03/18	Chinese Government Alters Threat Database Records posted on Recorded Future Blog
27/03/18	Soft Target: The Top 10 Vulnerabilities Used by Cybercriminals report released by Recoded Future
05/04/18	Mirai-Variant IoT Botnet Used to Target Financial Sector in January 2018 posted on Recorded Future Blog.
16/04/18	Integrating Threat Intelligence Into Security posted on Recorded Future Blog
25/04/18	North Korea's Ruling Elite Adapt Internet Behavior to Foreign Scrutiny posted on Recorded Futures Blog

Figure 4.6: Illustration of discovered topics from the Recorded Future dataset. The bottom panel shows posterior topic proportions timeline, solid bars represent topics and different colors represent different topics, topics that that inherit from previous epochs are connected with an area in the same color. The top panel shows the top ten words from the posterior distribution on the day that the topics appeared. Bottom panel lists posts from Recorded Future's Blog which likely might have triggered the topics discovered by the Model.

4.3 Hyperparameters μ and λ

In this section we do some exploration of the effects of the hyperparameters. The focus is on the two parameters μ and λ that are unique for the model, the hyperparameters γ and α are explored in other DP topics models. In order to explore how μ and λ affected the number of topics and the type to topics relations is discovered by the model we used the arXiv titles dataset. The model was tested using all combinations of $\mu = 0.01, 0.1, 0.2, \ldots, 0.9, 0.99$ and $\lambda = 0.01, 0.1, 0.2, \ldots, 1.0$, since the case $\lambda = 0$, $\mu = 0.0$ and $\mu = 1.0$ do not create a valid evolutionary topic model. In Figure 4.7 we see number of discovered topics, number of topics splits and number of new topics for each simulation. The numbers are extracted from epoch 4 (year 1997) from the dataset. The number of split topics in Figure 4.7(b) references to the number of topics whose parent has more than one inheritor in the current epoch.

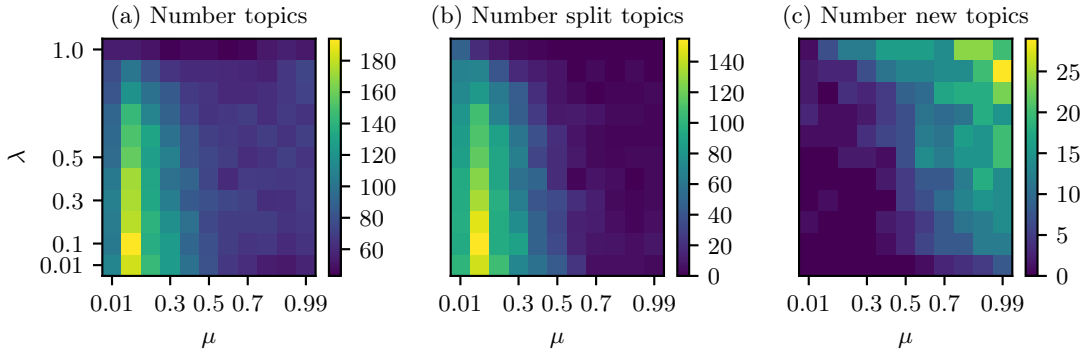


Figure 4.7: Effect of hyperparameters μ and λ on the (a) number of topics, (b) number topics split and (c) number of new topics. The numbers are gathered from the fourth epoch (1997) from the arXiv titles dataset. For μ values $0.01, 0.1, 0.2, \ldots, 0.9, 0.99$ were used and for λ values $0.01, 0.1, 0.2, \ldots, 1.0$, since the case $\lambda = 0$, $\mu = 0.0$ and $\mu = 1.0$ do not create a valid evolutionary topic model.

4.4 Stability and overfitting

To test the stability of the model we constructed a dummy dataset with the same observations for each epoch. The dataset consisted of titles from the arXiv dataset and had 1381 observations per epoch and repeated for 100 epochs. The dataset was preprocessed as described in Section 4.1 and after preprocessing the titles had an average length of 4.73 words. The dataset was split with 20 % of the data used as a test set for perplexity calculations (The same observations was used for testing in each epoch). In this evaluation the model was compared with the evolutionary topic model from [8], which creates an evolutionary DP by letting each topic create a DP in the next epoch. Since the model has a mixture of DPs we call it EMixDP. The hyperparameters used for EDPMM with hyperparameters $\gamma = 1.0$, $\alpha = 0.1$, $\mu = 0.5$ and $\lambda = 0.7$ and for EMixDP $\gamma = 1.0$, $\alpha = 0.1$, $\tau = 0.0001$, $\eta = 0.5$, $\lambda = 0.7$.

The perplexity and number of topics over time for the experiment is seen in Figure 4.8. In the perplexity calculation we see that both models has some overfitting but it is much worse for EMixDP. We also sees that EMixDP almost never decreases in the number of topics, instead it slowly increases over time. In the simulation, we used a low τ to limit the problem that the number of topics increasing over time, which is much worse for larger values of τ.

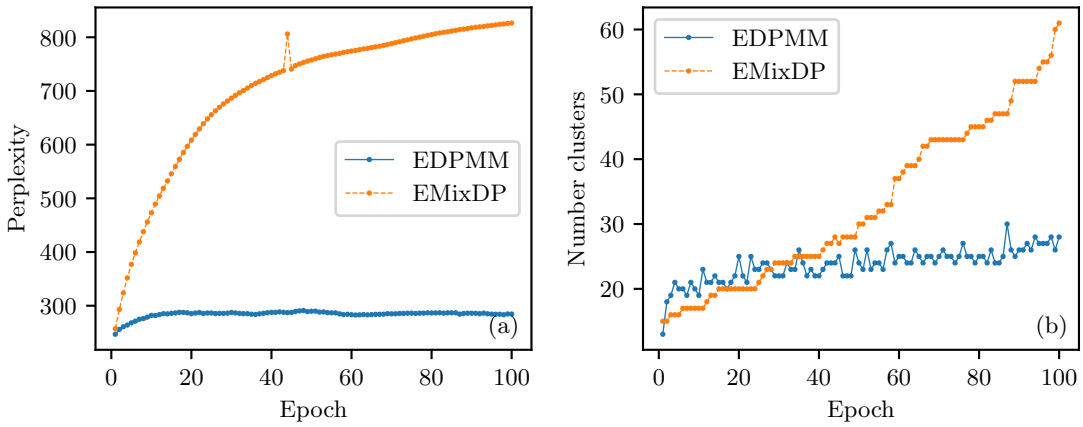


Figure 4.8: The perplexity and number of topics over time for a dummy dataset with the same observations each epoch. In the figure we see that EMixDP exhibits wore overfitting and not a stable number of topics.

Chapter 5

Discussion

This chapter includes a discussion and an interpretation of the results from the previous chapters. First the results from the two different datasets are discussed and interpreted. Then follows a discussion of the design of the model and how it is related to other similar DP topic models.

5.1 Topics discovered in the arXiv dataset

The arXiv dataset was used to show that EDPMM could discover long lived topics that possibly split into multiple topics. Example of discovered topics are shown in Figure 4.4 and Figure 4.5. The topics seem reasonable, but full evaluation of their connection to reality is hard. It also worth considering that they are only based on the titles of papers, so the information available to the model is limited.

5.1.1 Number of topics

The number of topics discovered for the EDPMM model in the arxiv growth over time is almost linear compared to the static topic models that exhibit more logarithmic growth, seen in Figure 4.3. The is interesting since all models put the same DP with a symmetric Dirichlet base distribution for the topics. One reason is since the EDPMM topics model allows for topics to split and therefore multiple topics to share the same prior, causing the effective number of observations used to be higher. Another contributing factor for this is in the probability of drawing a new topic in Gibbs sampling for EDPMM is dependent on topics in the previous epoch. The dependence is seen in Equation (3.11) the prior probability is multiplied by the likelihood of drawing that observation from the prior. For EDPMM the prior is the base prior combined with the prior from the topics in previous epochs. It is likely that one of the prior from the previous epoch assigns higher likelihood to an observation than the base symmetric Dirichlet prior, thus increasing the probability of drawing a new topic. This mechanism also helps the EDPMM model to converge faster then the static topic models since the prior gives a stronger indication of the expected topics.

The number of discovered topics for EDPMM is dependent on the hyperparameters used in the simulation. Even the hyperparameter μ and λ have a large unexpected impact on the number topics, this is discussed further in Section 5.5.

5.2 Topics discovered in Recorded Future dataset

The Recorded Future dataset was used to show EDPMM's ability to discover new topics in a short text data set. The topics discovered and related real events are shown in Figure 4.6. Several of the appearing topics with the most observations have a clear connection to blog post or other press releases made by Recorded Future. The observed data is the discussion on social media and articles written based on the original release by Recorded Future. Because our model recovers the topic evolution we can see that the discussion related to a release usually lasts a couple of days.

The two topics from dates 16/01/18 and 21/01/18 marked in Figure 4.6 are very similar but were not connected to each other. This might be because of the few number of observations on the day 20/01/18 in between. Since the model tries to recover statistical properties from the data is quite unstable when there is a small number of observations. This limitation of the model that once a topic has died it will not appear again might be a problem for certain use cases. A possible solution to this would be to let topics from several previous epochs affect the base measure of the DP. Allowing topics to have inheritors in more than one epoch. The number of future epochs a topic can have inheritors could either be a fixed number or dependent the size of the topic.

5.3 Dividing the data into epochs

The EDPMM topic model requires dividing the data into epochs of a fixed length before analysis. This division is arbitrary and often higher time resolution for the data is available. For the arXiv papers titles dataset we have the day the paper was published and for Recorded Future dataset we have exact hour or minute a text was published. Dividing up the data discards part of the data and is a limitation of the model. There are topic models for longer text from [16, 17] that are continuous time dynamic topic models. The documents are modeled as in LDA and the topic parameters are modeled as a stochastic processes and uses approximate variational inference. But since both [16] and [17] require specification of the number topics a priori, they do propose a solution for all problems.

5.4 Relation to other temporal DP topic models

There have been previous approaches of extending DPs to handle temporal data with applications in topic modeling. Early approaches in [18, 19] utilize dependant DP and do not allow for evolving cluster parameters over time. The topic model based on Recurrent Chinese Restaurant process (RCRP) from [20] does allow evolving cluster parameters. The RCRP model does allow for topics to appear and die out but it does not allow for a topic to split into multiple topics in the next epoch. The model from [8] where a Evolutionary DP is created by letting each epoch be a mixture of DPs. This approach allows both for evolution of topic parameters and for topics to split, the same as the model introduced in this thesis.

The biggest difference between our model and [8] is how the splitting of topics is modeled. In [8] it is directly controlled with the concentration parameters for the DPs based on a previous topic. This causes the model to often generate too many topics and not be as stable as our model. A second difference is that in our model we also sample the prior for topics in the Gibbs sampling algorithm. This allows the prior to a topic or multiple observations in one step, similar operation in [8] would require all observations to change to a new topic. This might help speed up the convergence of our model compared to the one in [8]. A last difference between the topic model in [8] and our model, is how we modeled the cluster parameters between epochs. In [8] they also applied the decay factor λ in Equation (3.8) on the base prior in addition to the observations. This sometimes causes topics with few observations to assign higher likelihood than the base distribution to unrelated observations. This can create topic connections and evolutions that are undesired and do not correspond to human interpretation. It also much worse then our model with respect to overfitting as shown in Figure 4.8(b).

5.5 Effect of hyperparameters

The interpretation of the hyperparameters has already been discussed in Subsection 3.4.3 and this chapter will cover observed effects of hyperparameters. The effect of μ and λ on the number of topics, number of topics splits and number of new topics is shown in Figure 4.7. Lower μ value the number of topic splits as expected from the interpretation of μ. The μ values also unexpectedly has a strong influence on the number of topics discovered. The higher λ gives more topics stronger memory, causing the model to discover more new topics instead of topics splits.

Optimization of the parameters with respect to for example perplexity can easily be done since both μ and λ fall into the range $[0, 1]$. The problem with optimizing the parameters is finding the proper value to optimize with respect to. Optimization with respect to perplexity might not give the most interpretable result. The values of the μ and λ as seen in Figure 4.7 have strong effect to the topic trees that are discovered. So the values might need to be tuned depending on what application the topic model is used for.

Chapter 6

Conclusions

6.1 Conclusions

In this thesis a non-parametric evolutionary topics model for short text capable of discovering topic births, deaths and splits is introduced. For inference a collapsed Gibbs sampling algorithm is derived. The model was empirically evaluated on real datasets to show it different abilities. On the first dataset the model's ability of discovering topic evolution and splitting over long time spans is shown. The second dataset shows the model's ability of discovering new topics and how the topics related to real life events is explored. We also showed that the model exhibits better stability and less overfitting than the short text topic model in [8] that has the ability to generate evolutionary topic trees.

6.2 Future work

6.2.1 Gibbs sampling computation time

Performing Gibbs sampling takes a long time for large datasets where the data contains many topics. It especially takes many iterations for the algorithm to converge fully which is seen in Figure 4.1. The parametric temporal topics models such as [5, 17] have used variational inference to archive fast a scalable inference. It is also possible to apply variational inference for DP mixture models [5]. In [5] they also showed that the variational inference achieves faster convergence than Gibbs sampling for certain types of data. Therefore variational inference might be a good way of decreasing the computation time needed for inference.

6.2.2 Tools for exploring topic models

Topic modeling is a good way of exploring and extracting information from large datasets. To take full advantage of the output from a topic model an interactive graphical inference is useful. For static topics models there is LDAvis [21] that is an interactive tool and inspired by this a similar tool for temporal topic models was implemented and used. There is still room for large improvements in this area, often topics are represented by just a few top words but this gives far from the full picture of the topic.

When working with large datasets that can contain over hundreds of topics it is still hard to get a good overview of the information. One way might be to use hierarchical modeling so there are different level of topics. This could be implemented directly into a topics model or by using a regular topic model. With a regular topic model it would first be used with hyperparameters so that it generates few topics and then using the topic model (with hyperparameters that gives more specific topics) again for the observations in each topic. This will make the exploration and understanding of large topics spaces more accessible.

Bibliography

[1] David M. Blei, Michael I. Jordan, Thomas L. Griffiths, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 17–24, Cambridge, MA, USA, 2003. MIT Press.

[2] Jianhua Yin and Jianyong Wang. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 233–242, New York, NY, USA, 2014. ACM.

[3] N.L. Hjort, C. Holmes, P. Müller, and S.G. Walker. *Bayesian Nonparametrics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010.

[4] Radford M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[5] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 113–120, New York, NY, USA, 2006. ACM.

[6] Yee Whye Teh. *Dirichlet Process*, pages 280–287. Springer US, Boston, MA, 2010.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[8] Peng Wang, Peng Zhang, Chuan Zhou, Zhao Li, and Hong Yang. Hierarchical evolving dirichlet processes for modeling nonlinear evolutionary traces in temporal data. *Data Mining and Knowledge Discovery*, 31(1):32–64, January 2017.

[9] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296, 2009.

[10] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 399–408, New York, NY, USA, 2015. ACM.

[11] DICE. Palmetto. `https://github.com/dice-group/Palmetto`, 2014.

[12] Cornell University. arXiv. `https://arxiv.org/`, 2018.

[13] Cornell University. arXiv, Open Archives Initiative API. `https://arxiv.org/help/oa/index`, 2018.

[14] Martin Porter. Snowball. `http://snowballstem.org/`, 2017.

[15] Recorded Future. Recorded future. `www.recordedfuture.com`, 2018.

[16] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, pages 579–586, Arlington, Virginia, United States, 2008. AUAI Press.

[17] Patrick Jähnichen, Florian Wenzel, Marius Kloft, and Stephan Mandt. Scalable generalized dynamic topic models. *arXiv preprint arXiv:1803.07868*, 2018.

[18] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Time-sensitive dirichlet process mixture models. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2005.

[19] Nathan Srebro and Sam Roweis. Time-varying topic models using dependent dirichlet processes. *UTML, TR# 2005*, 3, 2005.

[20] Amr Ahmed and Eric Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 219–230. SIAM, 2008.

[21] Carson Sievert and Kenneth Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70, 2014.

Appendix A

Dirichlet-Multinomial

We have observed data generated from a Multinomial distribution with a Dirichlet prior

$$\begin{aligned} \phi &\sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_V) \\ y &\sim \text{Multinomial}(\phi), \end{aligned} \tag{A.1}$$

with $y = (y_1, \ldots, y_V)$, $\phi = (\phi_1, \ldots, \phi_V)$ and $\sum_{i=1}^{V} y_i = n$. We want to calculate probability density function for the predictive posterior distribution called a Dirichlet-Multinomial distribution. The posterior predictive for new data $\tilde{y}$ is

$$p(\tilde{y}|y) = \int p(y|\phi)p(\phi)\,\mathrm{d}\phi = \int_{\Delta} n! \prod_{k=1}^{V} \frac{\phi_k^{y_k}}{y_k!} \frac{\Gamma(\sum_{i=1}^{V}\alpha_i)}{\prod_{k=1}^{V}\Gamma(\alpha_k)} \prod_{i=1}^{V} \phi_k^{\alpha_k - 1}\,\mathrm{d}\phi = \tag{A.2}$$

$$= \frac{n!\Gamma(\sum_{i=1}^{V}\alpha_i)}{\prod_{i=1}^{V} y_k!\Gamma(\alpha_k)} \int_{\Delta} \prod_{i=1}^{V} \phi_k^{\alpha_k - 1 + y_k}\,\mathrm{d}\phi, \tag{A.3}$$

here Δ is the V-simplex defined by $\sum_{i=1}^{V}\phi_i = 1$. By noticing that the integral in the last equation is normalizing constant for a Dirichlet distribution with parameters $\alpha_1 + y_1, \ldots, \alpha_V + y_V$ we have

$$\int_{\Delta} \prod_{i=1}^{V} \phi_k^{\alpha_k - 1 + y_k}\,\mathrm{d}\phi = \frac{\prod_{i=1}^{V}\Gamma(\alpha_k + y_k)}{\Gamma(\sum_{i=1}^{V}\alpha_i + y_i)} = \frac{\prod_{i=1}^{V}\Gamma(\alpha_k + y_k)}{\Gamma(n + \sum_{i=1}^{V}\alpha_i)} \tag{A.4}$$

Combining this with the previous result we have

$$p(\tilde{y}|y) = \frac{n!\Gamma(\sum_{i=1}^{V}\alpha_k)}{\prod_{i=1}^{V} y_k!\Gamma(\alpha_k)} \frac{\prod_{i=1}^{V}\Gamma(\alpha_k + y_k)}{\Gamma(n + \sum_{i=1}^{V}\alpha_k)} = \tag{A.5}$$

$$= \frac{n!\,\Gamma(\sum_{i=1}^{V}\alpha_k)}{\Gamma(n + \sum_{i=1}^{V}\alpha_k)} \prod_{k=1}^{V} \frac{\Gamma(y_k + \alpha_k)}{y_k!\,\Gamma(\alpha_k)} = \tag{A.6}$$

$$= \frac{n!}{\prod_{i=1}^{V} y_k!} \frac{\prod_{k=1}^{V}\prod_{i=1}^{y_k}(i - 1 + \alpha_k)}{\prod_{i=1}^{n}(i - 1 + \sum_{i=1}^{V}\alpha_k)}, \tag{A.7}$$

this is a closed form for the predictive posterior distribution.