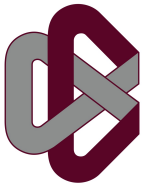
Centro de Investigación en Matemáticas



CIMAT

Hidden Markov Topic Models: Discrete Signal Analysis Using Markov Chain Monte Carlo

A dissertation submitted for the degree of

Master in Computer Science with specialization in Industrial Mathematics

Presents:

Fernando Fontove Herrera

Thesis advisor:

Dr. Salvador Ruiz Correa

Guanajuato, Guanajuato November 2011

I express my gratitude to CONACYT and CIMAT for providing the economic support to carry out my studies and this thesis.

To CIMAT because it has been an invaluable source of knowledge.

To my advisor, for his time and dedication.

And my deepest gratitude and love to my friends and family for their continuous support and help and for always being there to share my accomplishments.

Prologue

The human brain is a very complex system with rich spatio-temporal dynamics which are of great interest for research studies up to day. There exist several tools at our disposition to help explore the inner dynamics but the information provided by them hasn't been fully explored. Among the techniques for studying human brain dynamics, electroencephalography (EEG) is one of the most popular due to it being non-invasive and able to provide direct measures of the cortical activity with millisecond temporal resolution. The aim of this work is to create all the necessary software tools for implementing and testing some models for EEG signal analysis and studying the theoretic framework necessary for understanding and manipulating these models.

A state of the art technique used to analyse continuous EEG data is the continuous Hidden Markov Model (HMM) typically with the objective of classification and segmentation for applications that range from Brain Computer Interface to sleep EEG signals analysis[6]. There are many neurophysiological paradigms in which the information of interest is encoded in discrete-valued EEG signals (see [8]) and research with this kind of data is quite limited.

Before we introduce the discrete HMM we will talk briefly about a simpler model called Latent Dirichlet Allocation (LDA). In 1999 Hofmann presented the probabilistic Latent Semantic Indexing models (LSI). His idea was to represent a document as a mixture model, where the components are multinomial random variables that can be thought of as "topics". The problem with Hofmann's approach is that there is no generative probabilistic model for the mixing proportions of the components, which leads to some problems: the model grows linearly with the size of the corpus which leads to problems with overfitting; and it is not clear how to assign probability to a document outside of the training set [5].

The LDA model is a variation of Hofmann's LSI where a generative probabilistic model is fitted. Everything on the model is treated as a variable and priors are set upon them, in particular, the topics and the mixing proportions have a Dirichlet prior, hence the name.

It is important to remark that an assumption on which this method is based is that the document is simply a "bag-of-words", that is, the word order in a document can be neglected, making the variables are exchangeable [4] (as if the text was simply a bag with all the words in it). This assumption allows the use of de Finetti's representation theorem (1990), which states that any collection of exchangeable random variables has a representation as a mixture distribution. The goal of the LDA is to take advantage of this mixture representation and capture significant intra-document statistical structure [5].

The EEG data fails to satisfy the bag-of-words assumption as it clearly is time-dependent. For this reason, we will explore two variations of the LDA. The discrete HMM will take the basic generative probabilistic model from the LDA and make every latent state for each word depend on the previous state, creating a Hidden Markov chain (thus, discarding the bag-of-words assumption). This way, we partially keep the temporal relation between the data (which corresponds to grammar in the context of semantic analysis). The other variation we will explore is the Autoregressive Hidden Markov Model (AHMM), which pushes further the time dependency by making every latent state depend on the last two. Both of these models have been studied in the context of semantic analysis in Part-Of-Speech tagging and it is our belief that they are suited for the discrete EEG analysis.

Ultimately, the goal with these models is to find short descriptions of the members of a collection that enable efficient processing of large collections of data while preserving the essential statistical relationships for tasks such as classification and similarity and relevance judgements [5].

Once the generative model is set, a common approach is to maximize the probability of the hidden structure given the observed data. Typically this is done using maximum-likelihood estimation, a well studied algorithm for this is the expectation maximization (EM) algorithm, but, as it is well known, it doesn't perform well in high dimensionality as it easily gets stuck in local maxima [20]. For this reason, we will study sampling techniques to explore the parameter space and perform inference. We focus on the Metropolis-Hastings algorithm and variations from it (mainly the Gibbs sampler) as sampling algorithms and use simulated annealing to calculate the maximum

a posteriori (MAP) from the distribution. We will develop all the necessary software tools for implementing and testing the models.

The exposition is divided in 5 chapters. The first chapter contains the basic concepts needed and a very important result from Geyer that will help with the inference by providing window estimates. The second chapter is based on the book from Gilkins, Richardson and Spiegelhalter, “Markov Chain Monte Carlo in practice" and contains a study of the sampling algorithms we implement. On the third chapter we introduce the LDA, HMM and AHMM, also we discuss a problem that arises with these models called “label switching" and address how to deal with it. The fourth chapter contains several experiments we run to validate the models, the experiment with the EEG data and the tools we implemented to help with the data mining; we pay special attention to the AHMM as it shows the best results. Finally on the fifth chapter we explore some of the possible future work.

Fernando Fontove Herrera
Guanajuato, Gto, November 20 2011

Contents

Chapter 1

Generalities

We will introduce several concepts needed for the development of the models as well as the notation utilized throughout the work.

1.1 Bayesian Inference Paradigm

The problem we want to address is: given some data D, we want to do inference about some parameters θ that depend on D. Using the Bayes rule we know that

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}.$$

Since D is known, $P(D)$ is a constant and thus, just a scaling factor. In some applications, like the one in this work, suffice to have a function that is proportional to the probability via a constant; so the formula we will actually work with is

$$P(\theta|D) \ \propto \ P(D|X)P(\theta).$$

The term $P(D|\theta)$ is called the *likelihood*, $P(\theta)$ the *prior distribution* of θ and $P(\theta|D)$ the *posterior distribution* of θ. The goal of Bayesian inference is to compute a distribution over plausible parameter values, but often this distribution will not be analytically of numerically tractable [13]. For this reason we will study some sampling algorithms that will allow us to do inference.

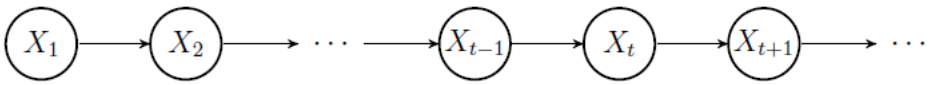


Figure 1.1: Graphical model for a Markov chain.

1.2 Markov Chains

Both the sampling algorithms and the models we explore have an underlying Markov chain on them, so we study the Markov chains briefly and enunciate some results we will need.

A *Markov chain* X is a collection of random variables $\{X_t\}_{t\in T}$, where T may be a finite or infinite set, that satisfies the relationship

$$P(X_t|X_1, X_2, ..., X_{t-1}) = P(X_t|X_{t-1}) \quad \text{for } t \in T,$$

that is, the current state depends only on the last one.

Here on, $P_{ij}(t)$ will be used to denote $P(X_t = j|X_1 = i)$ and

$$\tau_{ii} = \min\{t > 1 : X_t = i|X_1 = i\},$$

the first return to state i.

Definition 1.2.1. A Markov chain X is called

1. *Irreducible* if for all i, j there exists a $t > 0$ such that $P_{ij}(t) > 0$.
2. *Recurrent* if it is irreducible and $P(\tau_{ii} < \infty) = 1$ for some i.
3. *Positive recurrent* if $E(\tau_{ii}) < \infty$ for all i.
4. *Aperiodic* if it is irreducible and for all i,

$$\text{greatest common divisor}\{t > 0 : P_{ii}(t) > 0\} = 1.$$

Definition 1.2.2. Given a Markov chain X, $\pi(\cdot)$ is said to be a stationary distribution if for all j and $t \geq 0$

$$\sum_i \pi(i)P_{ij}(t) = \pi(j).$$

In plain terms, the last definitions ask that:

1. Any state may be reached from any starting state after some iterations.
2. The chain won't be oscillating between states at a fixed frequency.
3. If a sample drawn from X_t comes from the stationary distribution $\pi(\cdot)$ then all subsequent samples will also be distributed according to $\pi(\cdot)$.

The last property is the most important for us since it will allow the use of a Markov chain to obtain samples from any desired distribution.

Theorem 1.2.1. If a Markov chain X is positive recurrent and aperiodic then it has a unique stationary distribution $\pi(\cdot)$. Under these conditions X is called *ergodic* and satisfies

1. $P_{ij}(t) \to \pi(j)$ as $t \to \infty$ for all i, j.
2. If $E_\pi[f(X)] < \infty$ then

$$P(\bar{f}_N \to E_\pi[f(X)]) = 1 \text{ as } N \to \infty,$$

where

$$\bar{f}_N = \frac{1}{N} \sum_{n=1}^{N} f(x_n).$$

The first part of the theorem guarantees the convergence to the stationary distribution and the second gives an estimate for $E_\pi[f(X)]$, though it is important to notice it doesn't provide a way to determine how good it actually is. Something very important about this estimate is that the samples through which $\bar{f}_N$ is calculated need not be independent between them, as the law of big numbers would require.

Definition 1.2.3. A Markov chain is said to be *reversible* if it is positive recurrent, has a stationary distribution $\pi(\cdot)$ and satisfies

$$\pi(i)P_{ij} = \pi(j)P_{ij}.$$

This last property will be necessary to get a window estimate for $\bar{f}_N - E_\pi[f(X)]$, the error of the estimator.

1.3 Ergodic Averages

An inference problem we will adress later is to estimate the mean of a random variable given a set of samples from it. We will use the common estimator for the mean

$$\tilde{\mu} = \frac{1}{n}\sum_{n=1}^{N} x_n,$$

and we wish to know how good it is. The law of large numbers provides a partial solution by giving a window estimate, but requires independent samples which won't be possible with the sampling algorithms that we will use. The next theorem sidesteps this issue by using samples that come from a Markov chain.

Theorem 1.3.1. (Kipnis and Varadhan, 1986). For a stationary, irreducible, reversible Markov chain X and

$$\mu = \int g(x)dP(x), \qquad \tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} g(X_i), \qquad \gamma_t = \gamma_{-t} = Cov(g(X_i), g(X_{i+1})),$$

the following statement is true

$$nVar\tilde{\mu}_n \to \sigma^2 = \sum_{t=-\infty}^{\infty} \gamma_t \quad \textit{almost surely}$$

and if $\sigma^2 < \infty$, then

$$\sqrt{n}(\tilde{\mu}_n - \mu) \to N(0, \sigma^2).$$

It only remains for us to get an estimate for σ^2. There are a few considerations to make. First the Bartlett formula:

$$Var(\tilde{\gamma}_{n,t}) \approx \frac{1}{n}\sum_{s=-\infty}^{\infty} \gamma_s^2.$$

Second, there is no point summing terms where the autocovariance goes below the noise level inherent to the data, clearly it is wrong to add negative terms when we know they should be positive [3].

With these in mind we get the *initial positive sequence estimator* by stopping the summation at the first negative term m of

$$\tilde{\Gamma}_{n,m} = \tilde{\gamma}_{n,2m} + \tilde{\gamma}_{n,2m+1},$$

so we obtain

$$\tilde{\sigma}^2_{pos,n} = \tilde{\gamma}_0 + 2\sum_{i=1}^{2m+1} \tilde{\gamma}_{n,i} = -\tilde{\gamma}_0 + 2\sum_{i=0}^{m} \tilde{\Gamma}_{n,m}. \tag{1.1}$$

Finally the estimate we will use is

$$\tilde{\mu}_n - \mu \to N\left(0, \frac{\tilde{\sigma}^2_{pos,n}}{n}\right). \tag{1.2}$$

This theorem will allow us to extract samples from a Markov chain and use all of them to perform inference over the moments of a distribution without having to extract “semi-independent" samples. So if we want to calculate, say, the mean of the stationary distribution of a Markov chain, it suffices to average all the samples and the variance of the estimator will be given by (1.1) divided by the number of samples, as in (1.2).

1.4 Model selection

In general when a model is being fitted to some data, one has to take care how to select the model complexity, as simple models will fit poorly (underfit) and too complex models will overfit the data. This is easily seen when we get some points in the plane and try to fit a polynomial to them, if the degree is too low it will get a very poor fit and if it is large enough a perfect fit may be achieved, but the variance of the model will be very high with respect to the data, that is, slightly different points will give a very different polynomial [12].

It is important to keep in mind that in general it is not possible to get a “one true model" for the data. A model is a simplification of the reality, or as Box said, “All models are wrong, but some are useful"(1976). So we must not pursue the “true model", but an adequate enough model supported by the data and from which inference can be done [12, p.20 Sec. 1.2.5].

Usually model selection methods like Akaike's Information Criterion are used to determine which of several models is more accurate. The idea behind them is to penalize model complexity and balance under and overfitting [12].

Fortunately this issue will be solved implicitly thanks to the Bayesian approach used in this work. We provide a whole (parameterized) family of possible models, get the underlying distribution for them and sample from

it. So every sample we get will be a model for the data. Since the evidence is the probability that if parameters are randomly selected the data will be generated, simple models will be very unlikely to generate that particular data set. On the other hand, complex models can generate many possible data sets, so they are unlikely to generate that particular data set as well [2].

So in order to select which model is more adequate, we will rank them with respect to their likelihood and pick the highest one.

1.5 Topic Models

We will be working with the so called *topic models*, their objective is to extract the underlying *topics* from the data; that is, to find subsets of the data that are intra-related. In the context of semantic analysis, a topic would be a set of words that belong together, say articles, nouns or even a specific subject such as names or the terminology used in a more complex subject (for example in algorithm language: “for", “do",“while", etc.). One very important quality of these models is that they are able to discern the different meanings of homonyms, say, if “ruler" appeared in the text both as an instrument and as a monarch, it would appear in two different topics at the same time; this isn't possible with simple word counting algorithms.

Here on we will refer to every bit of the data as a *word*. Our approximation to these models is to assign a subjacent state to every word, so each state will have a set of words assigned to them and we can extract an associated probability distribution over the words for every state. When we refer to a topic we mean the state together with its corresponding distribution over the words. Formally, every word in the dictionary appears in a topic with a certain probability, but the probability of many of the words is so relatively small that it is negligible and it can be thought as if they didn't belong to the topic and it just consists of some of the words in the dictionary.

There are two very important aspects of the topic models: they offer a segmentation of the data and the topic themselves. Every word will have associated a topic that can be used to represent it and use it as a segmentation. So when the topics are extracted there are several uses: they can be analysed in order to find what is the connection between the words that the model is suggesting exists; and they provide a segmentation for the data that may be used either to compress it or ease its analysis by reducing the dimensionality.

1.6 Data presentation

Due to our intended application, we sort the data as follows: The whole data is the *corpus*, it consists of a group of *documents*, they are formed by *sentences* which in turn is composed by words. In the semantic context the meaning of the terms is intuitive, a corpus would be a complete book and its chapters are the documents. The EEG data we have consists of samples of people performing different tasks (mainly lexical decision, such as distinguishing if a word corresponds to an animal or an object), a task has different stimuli and we have the information from electrodes in different regions over the scalp. Then, the corpus for the EEG data will be the set of all the tasks plus the stimuli plus the regions, a document will be each of the possible combinations and a sentence is a sample of the experiment (i.e. the EEG signal corresponding to the response).

Defining the words will be the tricky part, first we must understand how the experiments were realized. We start by registering the EEG signal for a short period, then present the stimulus and continue registering until the response is presented (Fig. 1.2). Once we have the signal, a threshold based on the pre-stimulus period is used to determine wether the signal is activated or deactivated. Then it is discretized, so if the mean value of the signal over the discretized period is above the threshold it is represented with a '1', if it is equal a '0' and if it is lower a '-1'. This way, we turn an EEG signal into a string (sentence) with the words '0','1' and '-1'.

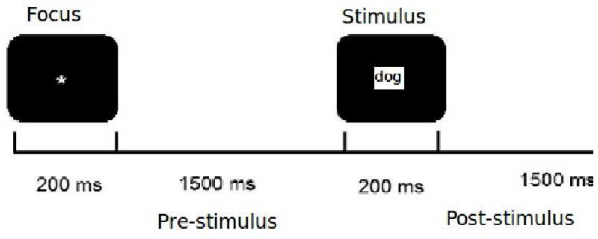


Figure 1.2: The pre-stimulus period has a duration of 1700ms, the stimulus is presented for 200ms and 1500ms are registered for the post-stimulus response.

There are several ways to go from here. We can use just the information of one frequency and electrode to be the words we will analyse or integrate them in some way. The decision of how to integrate the data will depend on the results we hope to find, in the experiment in this work we try to

determine the different uses of a specific region in the brain in two different tasks. For this reason we code the information as follows: At each period of the discretized time we have five different values of the signal corresponding to each of the frequencies (from 4 to 8) so we form a new dictionary with the numbers from 0 to $2^5 - 1$ where each represents a string formed by '0' and '1' (we discard the deactivation of the signal and take it as if there simply was no activation, i.e. a '0'). This way we get a dictionary with the words being the numbers 0, 1, ..., 31 and that is how we will represent the EEG data as text.

We use the topic models on the EEG data with the hopes that a topic will correspond to a set of patterns that belong to the same underlying process (for example reading, processing, motor response, etc). Also, we intend to use the segmentation as means to distinguish one experiment (document) from another in terms of the processes (topics) associated to them.

Chapter 2

Markov Chain Monte Carlo

As we said before, we will provide a parameterized family of models from which we wish to obtain samples. In general the distribution for this family is not tractable and special sampling algorithms are needed. With the aid of the theory on Chapter 1 we will construct a Markov chain that will allow us to extract samples from any desired distribution. The base algorithm that we will study is the *Metropolis Hastings* algorithm and the rest of the algorithms presented are some variations of it. The algorithms presented in this section are called *Markov Chain Monte Carlo* algorithms because of their use of an underlying Markov chain.

2.1 Metropolis Hastings

Our objective is to construct a Markov Chain that has our target distribution (for sampling) as its stationary distribution π. In order to do this, we get a sample X with the help of the previous sample Y and a distribution $q(X|Y)$ that is "easy" to sample together with the *acceptance function*

$$\alpha(X,Y) = \min\left(1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)}\right).$$

It can be shown that by drawing samples from any distribution $q(X|Y)$ and accepting them with probability $\alpha(X,Y)$, the Markov Chain obtained indeed has $\pi(\cdot)$ as its stationary distribution. This algorithm is called *Metropolis Hastings*.

Since every sample is drawn with help of the last, there is clearly a high correlation between them (this is the reason why the law of large numbers

Algorithm 1 Metropolis - Hastings

Initialize X^0 at random.
for $t = 1 :$ Number of Samples **do**
  Sample $Y \sim q(\cdot|X^t)$.
  Sample $u \sim U(0,1)$.
  if $u \leq \alpha(X_t, Y)$ **then**
    set $X^{t+1} = Y$.
  else
    set $X^{t+1} = X_t$.
  end if
end for

is of no use for us). Though this will not be a problem for us thanks to the results in section 1.3. There is a variation called *independence sampler* where $q(X|Y) = q(X)$, so every sample is independent from the others.

Single component Metropolis-Hastings. It is possible to divide a multidimensional variable X into several smaller components $\{X_1, ..., X_I\}$ of possibly differing dimensions and update them one by one. Notation: X_i^t will be the t^{th} sample of X_i and

$$X_{-i}^t = \{X_1^t, ..., X_{i-1}^t, X_{i+1}^{t-1}, ...X_I^{t-1}\}, \quad i = 1, ..., I.$$

The acceptance function takes the form

$$\alpha(X_i^t, Y_i) = \min\left(1, \frac{\pi(Y_i|X_{-i}^t)q_i(X_i^t|Y_i, X_{-i}^t)}{\pi(X_i|X_{-i})q_i(Y_i|X_i, X_{-i})}\right).$$

As mentioned before, we wish for our chain to be reversible. The Metropolis-Hastings algorithm gives a reversible chain by itself and in order to get a reversible chain in the single component version, it suffices to perform the updates in random order or set a fixed order for the variables and go over it forward and then backward each iteration [21].

2.2 Gibbs Sampler

The Gibbs Sampler is a special case of the *Single-component Metropolis - Hastings* in which the sampling distribution utilized $q_i(X_i|X_{-i})$ is in fact the

full conditional distribution of π i.e. $q_i(X_i|X_{-i}) = \pi(X_i|X_{-i})$. By sampling from the conditional distribution, the acceptance function is

$$\alpha(X, Y) = \min\left(1, \frac{\pi(Y)\pi(X|Y)}{\pi(X)\pi(Y|X)}\right) = \min\left(1, \frac{\pi(X, Y)}{\pi(X, Y)}\right) = 1,$$

so all the samples are accepted.

Algorithm 2 Gibbs Sampler

```
Initialize X^0 at random.
for t = 1 : Number of Samples do
  for i = 1 : I do
    X_i^t ~ π_i(·|X_{-i}^t).
  end for
end for
```

2.3 Modified Gibbs Sampler

Metropolis within Gibbs. It's not always possible to sample from the full conditional distributions, so sometimes it is necessary to use a metropolis step within the Gibbs sampler. Essentially, if the full conditional is available and easy to sample, it is used (Gibbs step), otherwise any other distribution is used and a Metropolis acceptance test is performed (Metropolis step).

Metropolized Gibbs Sampler. Liu proved that by using a *Random-Scan Metropolis-Hastings* with

$$q_i(Y_i \neq X_i^t|X_{-i}^t) = \frac{\pi(Y_i|X_{-i}^t)}{1 - \pi(X_i^t|X_{-i}^t)}$$

and

$$\alpha(X^t, Y_i) = \min\left(\frac{1 - \pi(X_i^t|X_{-i}^t)}{1 - \pi(Y_i|X_{-i}^t)}\right),$$

a more efficient chain is obtained in the sense that

$$\lim_{n\to\infty} \frac{1}{n}\text{var}\left(\sum_{t=1}^{n} f(X^t)\right)$$

is smaller, so the window estimate for $f(X)$ obtained from it is smaller[10, p.278 Sec. 13.3.1].

2.4 Convergence

We know that if samples are being drawn from a Markov chain, if one comes from the stationary distribution the subsequent will come as well from it. It is guaranteed that this will happen eventually thanks to theorem 1.2.1 when the chain is ergodic, which is the case when the Metropolis-Hastings algorithm is used [20]. Because of this, we will say a Markov chain has converged when the samples drawn from it come from the stationary distribution.

There is no infallible approach to check whether a chain has converged or not. Usually it may be "easily" determined just by inspection when convergence is achieved. For example on figure 2.1 we plot the log Posterior value of every sample on one of our experiments and one would guess convergence is reached at about 2000 iterations, where the distribution has apparently reached monotonicity; but intuition may deceive us and though figure 2.2 is the same as 2.1 but taking out the first 2000 samples, it is not as clear if convergence has been achieved.

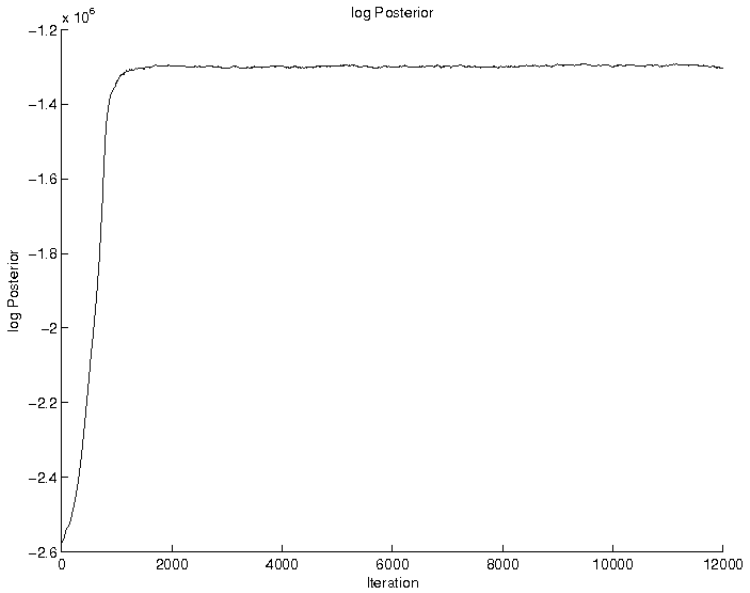


Figure 2.1: First 12000 samples of the logPosterior of a Gibbs sampler run. By inspection it appears to converge at about 2000 iterations.

The samples drawn before convergence is achieved are called the *burnin samples* and are discarded. Though, they need not be discarded for the

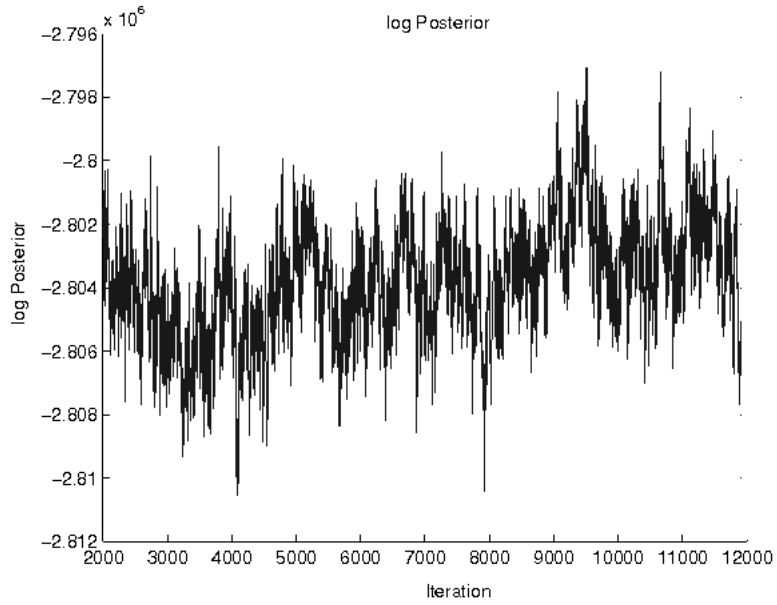


Figure 2.2: Same data as figure 2.1 but discarding the first 2000 samples and zooming in. The convergence is not as apparent.

inference, but if they are, the window estimate will be better. One common practice is to discard about 10% of the total chainlength [3].

We attempt to calculate the necessary burnin as follows: First of all, we assume the chain has converged and wish only to discard the burnin. Intuitively we think a chain has converged if it has "stabilized" in some sense, so we calculate the first and second moments of the tail of the chain and say the chain converged once the moments variation go bellow some threshold.

There is another more formal interpretation we may give to this algorithm. A distribution is completely determined by its moments, so if the moments associated to one population and another are the same, they both come from the same distribution [16, theo. 1 p. 208 sec. 8.3]. Parting from this idea, other methods may be implemented, such as hypothesis tests to check whether two populations come from the same distribution.

It is important to remember that the convergence should be checked for the parameters on which we want to perform the inference. So if we want to check the overall model convergence, we can watch for the convergence of the chain given by the log Posterior; but if we want to do inference over any

Algorithm 3 Burnin Calculation

$\mu^t = mean(X)$.
$\sigma^t = Var(X)$.
for $t = 1 : T$. **do**
  $\mu^t = mean(\{X_i\}_t^T)$.
  $\sigma^t = Var(\{X_i\}_t^T)$.
  if $\frac{\mu^t - \mu^{t-1}}{\mu^t} < \varepsilon$ and $\frac{\sigma^t - \sigma^{t-1}}{\sigma^t} < \varepsilon$. **then**
    $Burnin = t$.
    break.
  end if
end for

particular parameter, it should be checked on it as well.

Chapter 3

Latent Dirichlet Allocation and Hidden Markov Models

We will implement three different topic models with the objective of providing a segmentation for the data as means to analyse it and try to discriminate different classes. The core difference between the models is that on the first (Latent Dirichlet Alloction), the underlying structure assumes that there is no time dependency, contrary to the second (Hidden Markov Model) and third (Autoregressive Hidden Markov Model) where first and second order time dependencies are assumed on the data (every subjacent state depends on the previous or two previous ones).

Following the results of Griffiths and Goldwater [20], we will implement the *collapsed version* of these algorithms. We call a model "collapsed" when the parameters are integrated out. By integrating over the parameters, the model takes into account the consequences of possible variation in them, so greater robustness in the choice of state sequence is achieved [20].

3.1 Latent Dirichlet Allocation (LDA)

The Latent Dirichlet Allocation appears in the context of semantic analysis as a way to identify the subjacent topics in a given document. The document commonly is thought as a text but it may be any kind of data codified as words. The LDA recovers the topics and the overall form of the documents as mixtures of topics (a mixture distribution). As mentioned before, this model discards the time dependencies, so the order of the words in the document

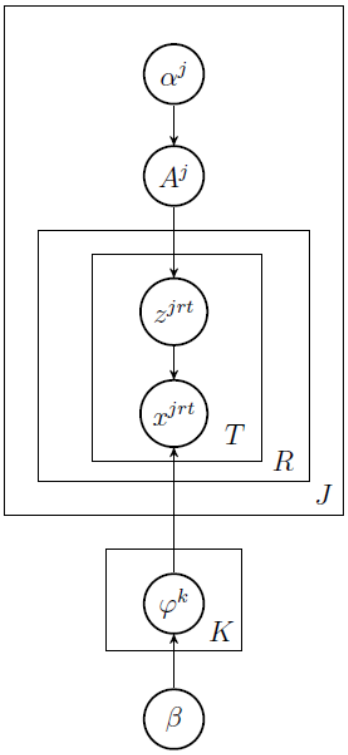


Figure 3.1: Graphical model for the LDA. J is the number of documents being analysed, K the proposed number of topics, R the number of texts in the document (can be thought as the chapters of the document) and T the length of the texts. The observed data x^{jrt} on the document is assigned a state z^{jrt}, the state depends on a multinomial distribution A and has associated a distribution ϕ_k respectively, they have α and β as concentration parameters for their a priori distributions.

is irrelevant for the analysis (in the context of semantic analysis, this can be interpreted as the grammar being discarded).

The states Z have a multinomial distribution A that has a Dirichlet prior with concentration parameter α. The data x has a distribution ϕ_k depending on the associated state z. The topics ϕ are distributed a priori as a Dirichlet with concentration parameter β.

This model is described in more detail in [5] and an implementation is provided in
`http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm`.

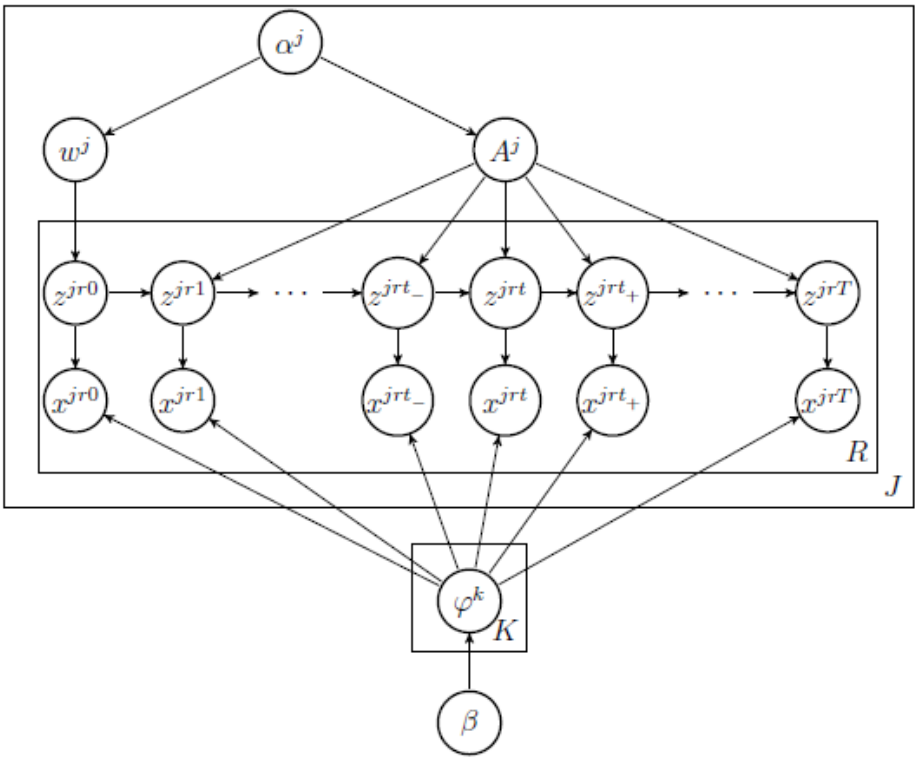


Figure 3.2: Graphical model for the HMM.

3.2 Hidden Markov Model (HMM)

The LDA model can be modified so it doesn't discard the time dependencies on the data. We do this by simply making every state depend on the last one, to obtain a discrete *Hidden Markov Model* (HMM). We set J as the number of documents, R the number of texts in the document, T the length of the texts and K the assumed number of topics. The transition matrix for the Markov Chain will be denoted as A_j and w_j the distribution for the initial state of the chain. Every state has a corresponding distribution φ_k.

A Uniform prior is given to the parameters α^j and β.

$$\alpha^j \sim \mathrm{U}(0,1).$$

$$\beta \sim \mathrm{U}(0,1).$$

The distribution w_j and every row of the transition matrices A_k^j have a Dirichlet prior with concentration parameter α.

$$w^j \sim \text{Dirichlet}(\alpha^j) \quad j \in J$$

$$A_k^j \sim \text{Dirichlet}(\alpha^j) \quad j \in J,\ k \in K.$$

The states z_{jrt} have a Multinomial prior.

$$z^{jr0} \sim \text{Mult}(w^j) \quad j \in J,\ r \in R.$$

$$z^{jrt} \sim \text{Mult}(A^j_{z_{jr(t-1)}}) \quad j \in J,\ r \in R,\ t > 0.$$

The topics φ^k have a Dirichlet prior with concentration parameter β.

$$\varphi^k \sim \text{Dirichlet}(\beta) \quad k \in K.$$

The generative model is then:

Algorithm 4 Ancestral Sampling for the HMM

sample $\alpha \sim U(0,1)$.
sample $\beta \sim U(0,1)$.
for $k = 1 : K$ **do**
  sample $\varphi^k \sim \text{Dirichlet}(\beta)$
end for
for $j = 1 : J$ **do**
  sample $w^j \sim \text{Dirichlet}(\alpha)$
  for $k = 1 : K$ **do**
    sample $(A^j)_k \sim \text{Dirichlet}(\alpha)$
  end for
  for $r = 1 : R$ **do**
    sample $z^{jr0} \sim \text{Multinomial}(w_j)$
    for $t = 1 : T$ **do**
      sample $z^{jrt} \sim \text{Multinomial}(A^j_{z_{jr(t-1)}})$
    end for
    for $t = 0 : T$ **do**
      sample $x_{jrt} \sim \text{Multinomial}(\varphi^{z_{jrt}})$
    end for
  end for
end for

From the graphical model we obtain the following factorization of the joint distribution

$$\begin{aligned} P(z,x,A,w,\varphi|\alpha,\beta,\gamma) &= \prod_{j=1}^{J} P(\alpha^j)P(A^j|\alpha)P(w^j|\gamma) \\ &\quad \prod_{j=1}^{J}\prod_{r=1}^{R} P(z_{jr0}|w_j)\prod_{t=1}^{T} P(z_{jrt}|z_{jr(t-1)},A_j) \\ &\quad \prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=0}^{T} P(x_{jrt}|z_{jrt},\varphi) \\ &\quad P(\beta)\prod_{k=1}^{K} P(\varphi_k|\beta). \end{aligned}$$

For the collapsed model, the variables φ, A^j and w^j are marginalized to obtain

$$\begin{aligned} P(z_{jrt,s}=1|-) &= \frac{\nu_{sw}+\beta}{\nu_{s\cdot}+W_s\beta}\frac{\eta_{jks}+\alpha}{\eta_{ck\cdot}+K\alpha} \\ &\quad \frac{\eta_{csr}+I(k=s)I(r=s)+\alpha}{\eta_{cs\cdot}+I(k=s)+K\alpha}, \end{aligned}$$

where η_{jks} is the number of times in experiment J that state s is obtained after state k, $\eta_{jk\cdot}$ the number of times in experiment J that state k appears, ν_{sw} the times state s appears together with word w and $\nu_{s\cdot}$ the total number of times state s appears.

For the non-collapsed model, the conditional probabilities are

- $P(\alpha^j|-) \propto U(0,1)\left[\prod_{k=1}^{K}\prod_{l=1}^{K}(A^j_{kl})^{\alpha-1}\right]\prod_{l=1}^{K}(w^j_l)^{\alpha-1}$.
- $P(w^j|-) \propto Dir((\sum_{r=1}^{R} z_{jr0,1},\sum_{r=1}^{R} z_{jr0,2},...,\sum_{r=1}^{R} z_{jr0,K})+\alpha^j)$.
- $P(A^j_k|-) \propto Dir\left(\alpha^j+\left(\sum_{r=1}^{R}\sum_{t=1,z_{jr(t-1)}=k}^{T} z_{jrt,l}\right)_{l=1}^{K}\right)$.
- $P(z_{jr0}|-) \propto \prod_{l=1}^{K}(w^j_l)^{z_{jr0,l}}\prod_{l=1}^{K}\prod_{w=1}^{W}(\varphi^l_w)^{z_{jr0,l}x_{jr0,w}}$ $\prod_{k=1}^{K}\prod_{l=1}^{K}(A^j_{kl})^{z_{jr1,l}z_{jr0,k}}$.

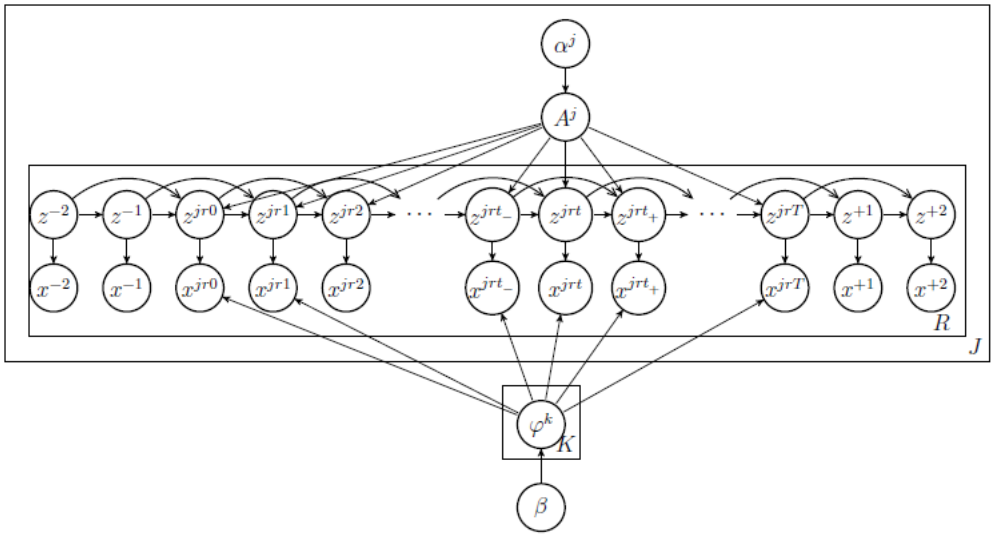


Figure 3.3: Graphical model for the AHMM.

- $P(z_{jrt}|-) \propto \prod_{l=1}^{K}\prod_{w=1}^{W}(\varphi_w^l)^{z_{jrt,l}x_{jrt,w}}\prod_{k=1}^{K}\prod_{l=1}^{K}(A_{kl}^j)^{z_{jrt,l}z_{jr(t-1),k}}$ $\prod_{k=1}^{K}\prod_{l=1}^{K}(A_{kl}^j)^{z_{jr(t+1),l}z_{jrt,k}}$.
- $P(z_{jrT}|-) \propto \prod_{k=1}^{K}\prod_{l=1}^{K}(A_{kl}^j)^{z_{jrT,l}z_{jr(T-1),k}}\prod_{l=1}^{K}\prod_{w=1}^{W}(\varphi_w^l)^{z_{jrT,l}x_{jrT,w}}$.
- $P(\varphi^k|-) \propto Dir(\beta + (\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k}^{T} x_{jrt,1}, ...,$ $\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k}^{T} x_{jrt,W}))$.
- $P(\beta|-) \propto U(0,1)\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}$.

3.3 Autoregressive Hidden Markov Model (AHMM)

We now push the time dependencies further on the LDA model by making every state depend on the last two and call it *Autoregressive Hidden Markov Model*. Now A_j is a the transition cube for the autoregressive Markov chain. Dummy data are inserted at the beginning and end of the data as to avoid

special cases where a variable has one or none ancestors, so, $z_{jr-2} = z_{jr-1} = z_{jr+1} = z_{jr+2} = 0$ and $x_{jr-2},\ x_{jr-1},\ x_{jr+1},\ x_{jr+2}$ will be the empty word "".

Similar to the Hidden Markov Model, the priors are:

- $\alpha \sim U(0,1)$.
- $\beta \sim U(0,1)$.
- $A^j_{k,l} \sim Dirichlet(\alpha)\ \ j \in J,\ k,\ l \in K$.
- $z^{jrt} \sim Mult(A^j_{z_{jr(t-2)},z_{jr(t-1)}})\ \ j \in J,\ r \in R$.
- $\varphi^k \sim Dirichlet(\beta)\ \ k \in K$.

Note that now $A^j_{k,l,m}$ represents the probability of getting to state m when $z_{jr(t-1)} = l$ and $z_{jr(t-2)} = k$.

From the graphical model we obtain the following factorization of the joint distribution

$$\begin{aligned} P(\mathbf{z}, \mathbf{x}, \mathbf{A}, \alpha, \beta, \varphi) &= P(\beta)\left[\prod_{k=1}^{K} P(\varphi^k|\beta)\right] \\ &\prod_{j=1}^{J} P(\alpha^j)P(A^j|\alpha^j) \\ &\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T} P(z_{jrt}|z_{jr(t-2)}, z_{jr(t-1)}, A^j)P(x_{jrt}|z_{jrt}, \varphi) \end{aligned}$$

For the collapsed model, the variables φ, A^j and w^j are marginalized to obtain

$$\begin{aligned} &P(z_{jrt,k} = 1|-) = \\ &\frac{\nu_{s,w} + \beta}{\nu_{s\cdot} + W_s\beta}\frac{\eta_{j,p_2,p_1,k} + \alpha}{\eta_{j,p_2,p_1} + K\alpha}\frac{\eta_{j,p_1,k,n_1} + I(p_2 = p_1 = k = n_1) + \alpha}{\eta_{j,p_1,k} + I(p_2 = p_1 = k) + K\alpha} \\ &\frac{\eta_{j,k,n_1,n_2} + I(p_2 = k = n_2, p_1 = n_1) + I(p_1 = k = n_1 = n_2) + \alpha}{\eta_{j,k,n_1} + I(p_2 = k, p_1 = n_1) + I(p_1 = k = n_1) + K\alpha}, \end{aligned}$$

where $\eta_{j,a,b,c}$ is the number of times in experiment J that state c is obtained after state b after a; $\eta_{jab\cdot}$ the number of times in experiment J that

state b appears after a; ν_{sw} the times state s appears together with word w; $\nu_{s\cdot}$ the total number of times state s appears; $p_2 = t-2$, $p_1 = t-1$, $n_1 = t+1$ and $n_2 = t+2$.

For the non-collapsed model, the conditional probabilities are

- $P(\alpha^j|-) \propto U(0,1)\prod_{k=1}^{K}\prod_{l=1}^{K}\prod_{m=1}^{K}(A^j_{klm})^{\alpha^j-1}$.
- $P(A^j_{kl}|-) \propto Dir(\alpha^j + \left(\sum_{r=1}^{R}\sum_{t=1}^{T} z_{jrt,m} I(z_{jr(t-2)}=k,\ z_{jr(t-1)}=l)\right)_{m=1}^{K})$.
- $P(z_{jrt}|-) \propto A^j_{p_2,p_1,k}A^j_{p_1,k,n_1}A^j_{k,n_1,n_2}\varphi_y^{z_{jrt}}$.
- $P(\varphi^k|-) \propto Dir\left(\beta + \left(\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k}^{T} x_{jrt,m}\right)_{w=1}^{W}\right)$.
- $P(\beta|-) \propto U(0,1)\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi^k_w)^{\beta-1}$.

3.4 Sampling the models

The HMM and the AHMM will be implemented using a metropolis-within-gibbs sampler. For both of them the collapsed form allows a Gibbs sampler step for the state variables z and for the hyper parameters α and β a metropolis step is used [20].

The models will be run checking for convergence with the tools developed in section 2.4 using the samples from the log-posterior.

3.5 Label Switching

A troubling issue that will arise while sampling is that the states won't be identifiable, that is, if the labels assigned to them are permuted, the samples are "essentially the same". In more precise terms, since exchangeable priors are placed upon the parameters on the models we use (this is the case as all of them have the same prior), the resulting posterior distribution will be invariant under permutations of the parameter labels. This will invalidate doing inference on some parameters by simply averaging over the samples [1].

In general, label exchangeability implies that in a K component mixture, the number of modes on the posterior is of order $O(K!)$ and though with the HMM and AHMM we don't have a mixture model as on the LDA, they behave very much like one on this aspect. A naïve solution is to impose identifiability constraint on the parameters, but this will truncate the posterior and it won't agree with the original (which is the one we actually want to sample from) [11].

The approach we will use is to relabel the samples after obtaining all of them. This is essentially a way to force identifiability without altering the original distribution (as the samples have already been obtained from it). To do it we stablish a "reference" sample, which will be the maximum a posteriori (MAP) and relabel all the other samples by "aligning" them with it: a variable will be assigned the label of the closest one to it from the reference sample under some distance. In our case the topics labels are the ones that suffer from label switching, since they are distributions we will use the *Hellinger distance* [7] as a similarity measure.

There is also a way to sidestep this issue: we can focus on quantities that are invariant under the permutations, for example the value of the likelihood or the posterior are not affected. As with the convergence, it is important to consider the aim of the inference, since it could very well happen that the label switching doesn't have an effect on the analysis.

Chapter 4

Experiments

We will validate the models by means of experiments with synthetic data. First we will generate samples with the model assumed and then try to recover it. For the second part we generate artificial data similar to what we see (in an exploratory analysis) on the EEG data and try to recover the underlying states. For the last part we use the autoregressive model to explore the real EEG data.

4.1 HMM synthetic experiment

Data samples are generated using 10 topics each with 25 words, 2 different classes (transition matrices) and 2000 repetitions (samples) are obtained with a length in time of 200.

We will use the topics described in figure 4.1 to generate the data because they have an easy representation as an image which will allow us to check easily if they are recovered correctly.

The transition matrices are generated using a Dirichlet distribution with concentration parameter $\alpha = .7$.

A Gibbs sampler is used for the model described in section 3.3 fixing the

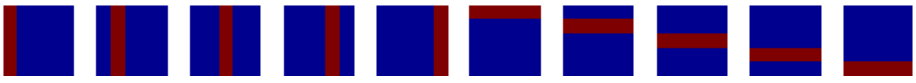


Figure 4.1: Topics used. Each square represents a distribution over the words, a blue coloring in a pixel means that topic has low probability on that word and red is for high.

Algorithm 5 Data generation

for $j = 1 : J$ **do**
  for $r = 1 : R$ **do**
    sample $z^{jr1} \sim U(1, ..., K)$
    for $t = 2 : T$ **do**
      sample $z^{jrt} \sim Multinomial(A^{j}_{z_{jr(t-1)}})$
    end for
    for $t = 0 : T$ **do**
      sample $x_{jrt} \sim Multinomial(\varphi^{z_{jrt}})$
    end for
  end for
end for

topics from 2 to 15 and run till convergence is achieved under the criteria mentioned in section 2.4.

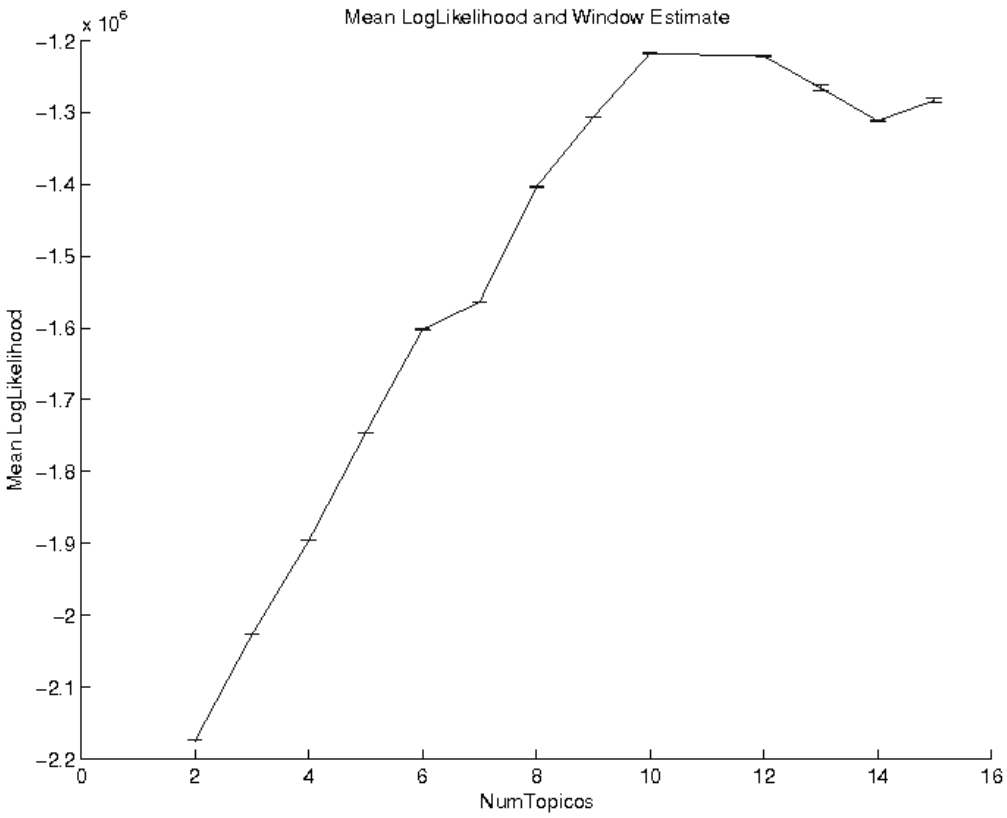


Figure 4.2: Mean log likelihood for the sampled topics with the standard deviation for the estimator as error bars.

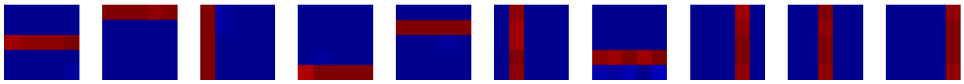


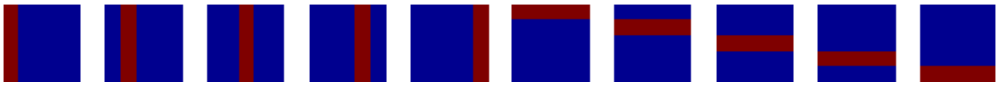
Figure 4.3: Recovered topics when trying to fit 10.

4.2 Autoregressive

The first three experiments are done in order to validate the model, an autoregressive model is used to generate the data and we try to recover it. The fourth and fifth experiments try to simulate real data. The model selection tools implemented are shown on the first four experiments and the data analysis tools on the last experiment.

4.2.1 Synthetic experiment 1

The data is generated using 10 topics each with 25 words, 2 different classes (transition matrices) and 2000 repetitions are obtained with a length in time of 200. The topics used are:



Each square represents a distribution over the words. A blue coloring in a pixel means that topic has low probability on that word and red is for high. These topics are used because they have an easy representation as an image which will allow us to check easily if they are recovered correctly.

The transition matrices are generated using a Dirichlet distribution with concentration parameter $\alpha = .7$.

Algorithm 6 Data generation

for $j = 1 : J$ **do**
  for $r = 1 : R$ **do**
    sample $z^{jr1} \sim U(1, ..., K)$
    sample $z^{jr2} \sim U(1, ..., K)$
    for $t = 3 : T$ **do**
      sample $z^{jrt} \sim \text{Multinomial}(A^{j}_{z_{jr(t-2)}, z_{jr(t-1)}})$
    end for
    for $t = 0 : T$ **do**
      sample $x_{jrt} \sim \text{Multinomial}(\varphi^{z_{jrt}})$
    end for
  end for
end for

A Gibbs sampler for the model described in section 3.3 fixing the topics from 1 to 14 and run till convergence is achieved under the criteria mentioned in section 2.4.

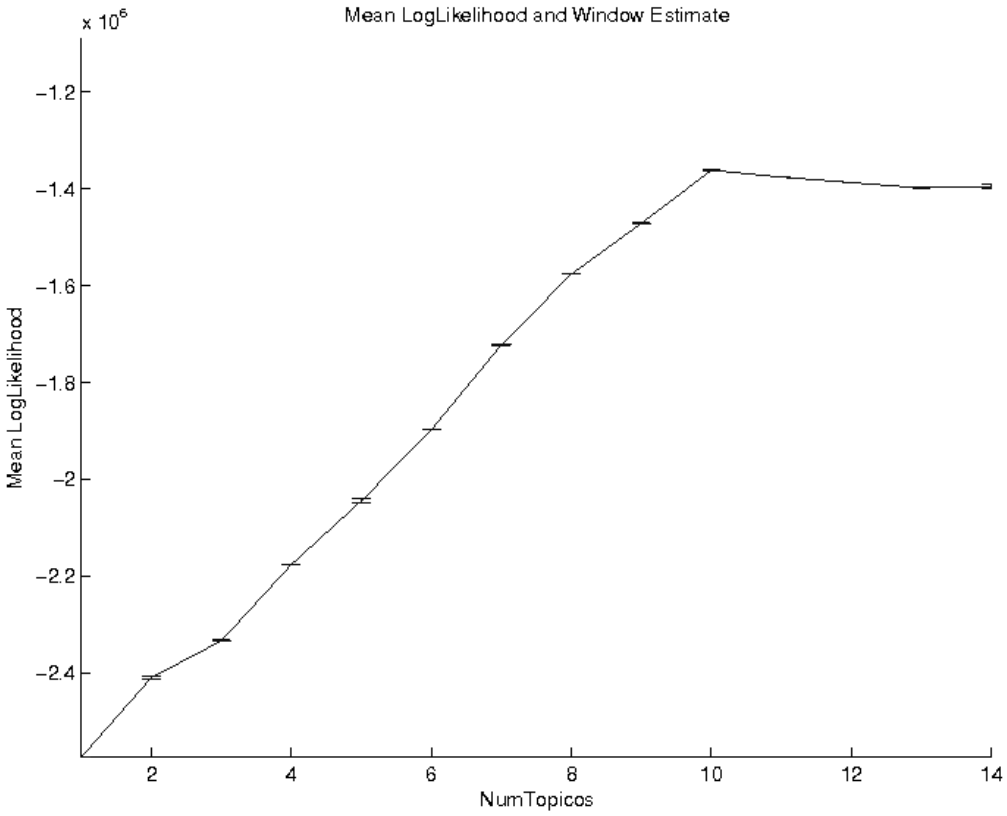


Figure 4.4: Mean log likelihood for the sampled topics with standard deviation of the estimator as error bar.

Discussion

The maximum a posteriori (MAP) and maximum likelihood are both achieved at 10 topics which coincides with how the data was generated. The topics are recovered successfully but as mentioned on section 3.3, the model suffers from label switching and we obtain a permutation of the original topics.

Even though the MAP is achieved at 10 topics model, hence is the most supported by the data, the difference in posterior from 10 to 15 topics is not

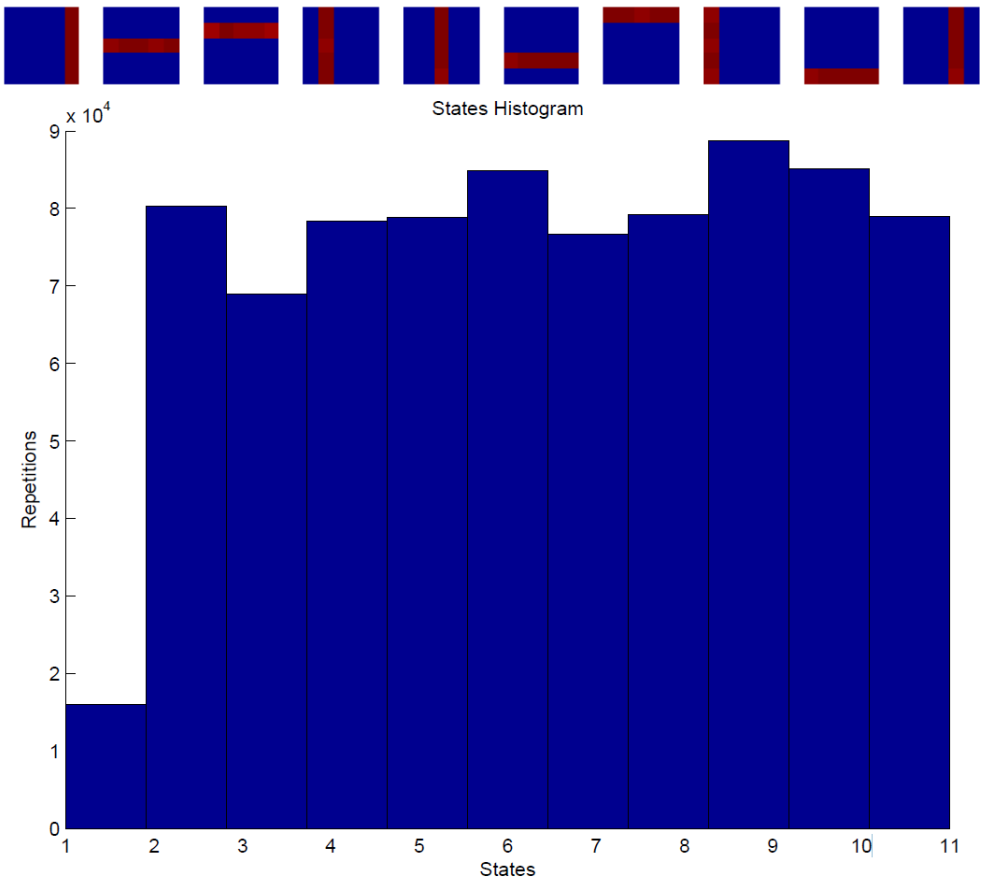


Figure 4.5: Recovered topics with a histogram of their appearance.

big. Doing a deeper analysis of these topics we find that the 10 “true" topics are recovered as well when 11 or more topics are fitted and the extra are spurious topics. By checking a histogram of the overall appearances of the topics on the sample we can see that these spurious topics have a relative very low appearance compared with the true ones. This also supports the hypothesis that the 10 topic model is the best one.

Usually the spurious topics are a dummy topic with extremely low relative appearance on the data. Other common thing that happens when more topics than necessary are fitted is that a big topic is split into several copies.

NumTopicos	mLogLikelihood	logMAP	Estimator error
2	-2.574592e+06	-2.599711e+06	5.487687e+01
3	-2.410639e+06	-2.969761e+06	5.237675e+06
4	-2.334129e+06	-3.146129e+06	1.206159e+06
5	-2.176384e+06	-3.119782e+06	1.089212e+06
6	-2.044888e+06	-3.169834e+06	1.872474e+07
7	-1.897656e+06	-3.138354e+06	3.115666e+05
8	-1.722277e+06	-3.041400e+06	1.249954e+06
9	-1.575935e+06	-3.006190e+06	2.317599e+05
10	-1.470828e+06	-2.935541e+06	1.216239e+05
11	-1.360996e+06	-2.859228e+06	1.444285e+05
12	-1.375521e+06	-2.892284e+06	3.019420e+05
13	-1.386613e+06	-2.910322e+06	1.816601e+05
14	-1.397859e+06	-2.921394e+06	4.100085e+05
15	-1.394386e+06	-2.944958e+06	1.537178e+07

Figure 4.6: Results summary.

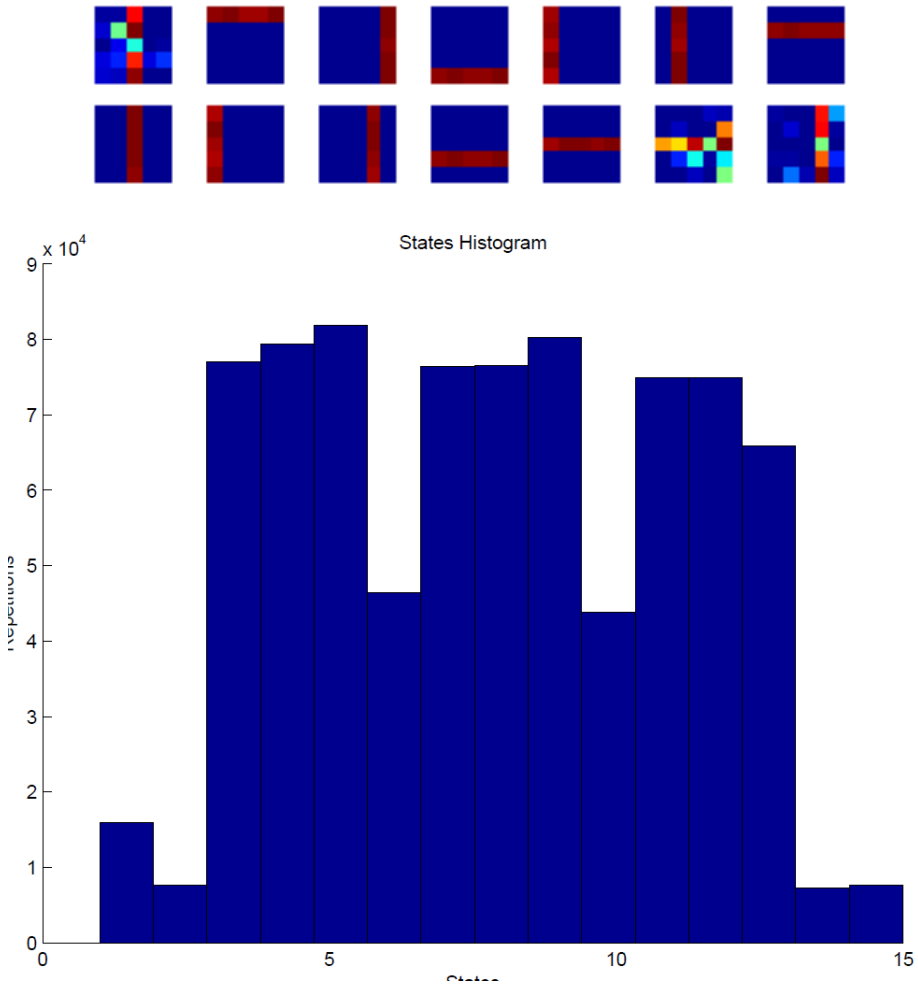


Figure 4.7: Fitting 14 topics we can see how the topics 2, 13 and 14 which are spurious, actually have very few appearances over the data.

4.3 Synthetic experiment 2

The data is generated the same way as in experiment 1 with the difference that the probability to go to state 1 from any other state is greater than .3. By forcing this, the state 1 takes a big part of the mixture distribution and "overshadows" the other ones. We do this to exemplify that even though the states have a relatively low appearance, the algorithm still recovers them successfully.

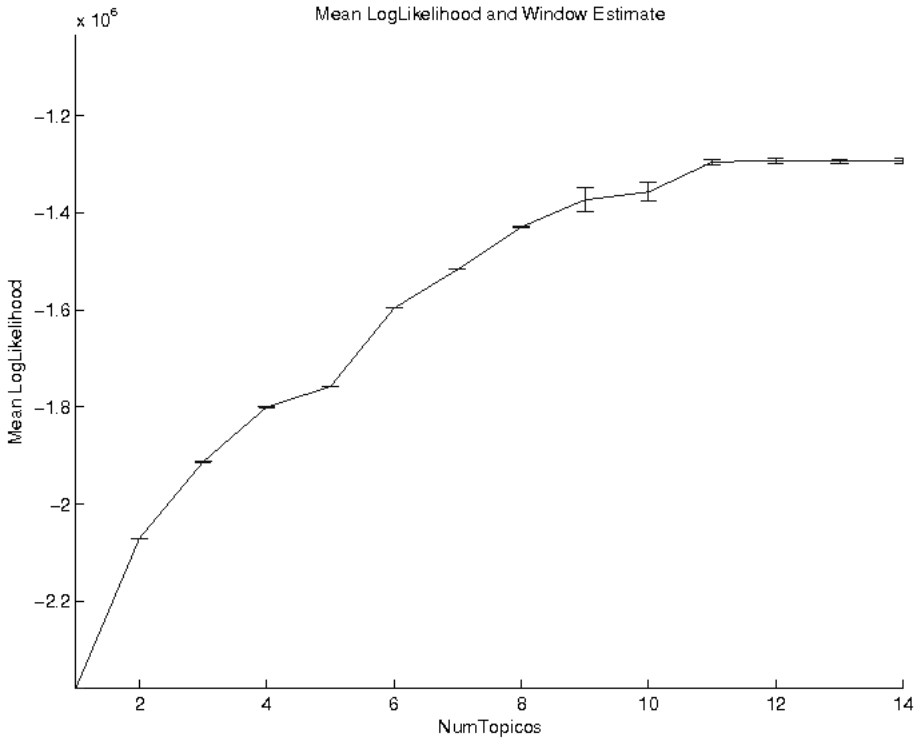


Figure 4.8: Mean log likelihood for the sampled topics with the standard deviation of the estimator as error bar.

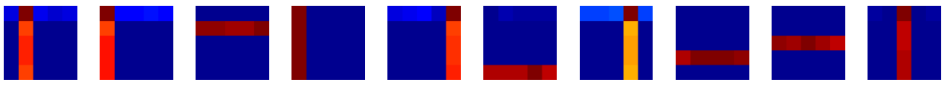


Figure 4.9: Recovered topics.

NumTopicos	mLogLikelihood	logMAP	Estimator error
2	-2.379336e+06	-2.404493e+06	6.694947e+01
3	-2.070403e+06	-2.613315e+06	8.950286e+04
4	-1.912947e+06	-2.678430e+06	1.869360e+05
5	-1.800185e+06	-2.684018e+06	2.588052e+05
6	-1.758056e+06	-2.754567e+06	6.373533e+05
7	-1.596566e+06	-2.696563e+06	1.245278e+05
8	-1.516764e+06	-2.657337e+06	2.051389e+05
9	-1.429136e+06	-2.620923e+06	3.451381e+04
10	-1.373566e+06	-2.624223e+06	6.441351e+08
11	-1.357282e+06	-2.719183e+06	3.857884e+08
12	-1.295566e+06	-2.697561e+06	2.453540e+07
13	-1.292863e+06	-2.750091e+06	2.181223e+07
14	-1.294268e+06	-2.743427e+06	2.164255e+07
15	-1.293192e+06	-2.819854e+06	4.129407e+07

Figure 4.10: Results summary.

4.3.1 Discussion

The results are very similar as the ones on the first experiment. The maximum is achieved when all the original topics are recovered even when the distribution is largely dominated by only one state. As with the first experiment, after all the topics are recovered one of two things occur, an extra topic is a copy of one of the originals or it is filled with “garbage" and has a very low appearance.

4.3.2 Unsuccessful experiment

When the data is generated with a parameter of $\alpha = 10$ or bigger, the algorithm is not able to recover the topics in the tests run of up to $30,000$ samples (usually 1000 are enough). This is not as bad as it would appear since by having such a high value for the concentration parameter of a Dirichlet distribution the distributions sampled from it will have extremely high entropy, therefore the change from one state to another is almost random with an uniform distribution and the essential structure of the model is lost. In principle, the model would still be present but due to the limited computer precision, it is lost.

4.4 Synthetic experiment 3

The data is generated by imitating the sort of data we see on the EEG images: once a state is achieved it prevails for a short period and then jumps to another. For a repetition, a state is selected randomly from the twelve available and its duration is sampled from a uniform distribution. Since the states are picked randomly, it is as if every sample belonged to a different class.

After the states are determined the topics from experiment 1 are used again together with two new states. The first gives a probability of .7 to the first word and is uniform among the remaining words. The second gives a probability of .4 to the first and second words and is uniform among the remaining words.

4.4.1 Discussion

Though the data is not generated using an underlying autoregressive model per se, the topics are successfully recovered. Unfortunately, the maximum is not achieved at 12 topics as it should. This issue has to be addressed with care, since this algorithms are probabilistic in nature, they may not achieve the desired result and two things need to be done: if possible, run several times to get some degree of assurance that the result is not a local optima and second, the results have to be checked manually. Checking them manually implies that we have to see if the results actually make sense to the data we are trying to fit and try to find any sort of anomaly.

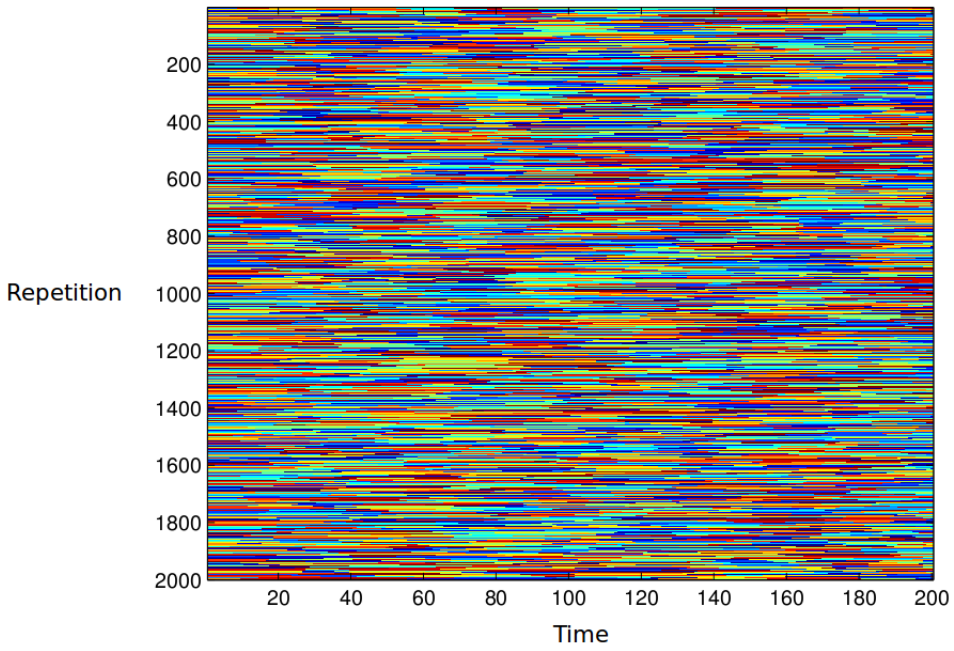


Figure 4.11: Visualization of the states used to generate the data. Each color represents a different state.

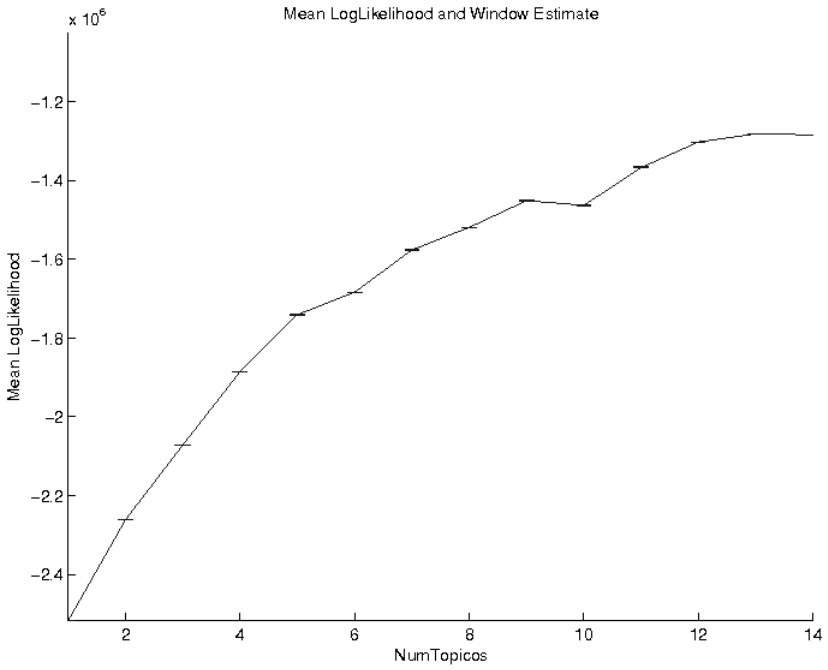


Figure 4.12: Mean log likelihood for the sampled topics with the standard deviation of the estimator as error bar.

Figure 4.13: Recovered topics.

As mentioned before, the two anomalies we commonly found in data are: split topics and spurious topics with extremely low appearance. If we check closely the results, we can see with the help of the topic distance matrix on figure 4.15 that one of the topics is repeated when trying to fit the 12 topics. Also if we check with 14 topics, one of them is repeated as well and there are 12 different topics with significant appearances as shown by figure 4.16. This has to be treated with care, naively we could just merge the split topics, but in theory, it may very well be that the reason for the split is the time dependency and that even though they look the same, they are in fact, different.

In a second run of this data, the topics are successfully recovered when trying to fit 12 topics.

NumTopicos	mLogLikelihood	logMAP	Estimator error
2	-2.516858e+06	-2.542010e+06	1.729582e+01
3	-2.260895e+06	-2.371946e+06	4.642837e+03
4	-2.071217e+06	-2.233932e+06	4.982513e+04
5	-1.885132e+06	-2.069785e+06	1.536878e+04
6	-1.740676e+06	-1.940174e+06	3.641916e+03
7	-1.683423e+06	-1.895576e+06	1.904117e+04
8	-1.576213e+06	-1.799851e+06	1.226333e+04
9	-1.518912e+06	-1.751936e+06	1.139804e+04
10	-1.451017e+06	-1.695832e+06	2.333702e+05
11	-1.463065e+06	-1.722366e+06	1.458959e+05
12	-1.366932e+06	-1.629315e+06	1.468031e+06
13	-1.302087e+06	-1.574107e+06	1.808109e+05
14	-1.281564e+06	-1.566342e+06	7.435983e+04
15	-1.283979e+06	-1.570829e+06	7.042044e+04

Figure 4.14: Results summary.

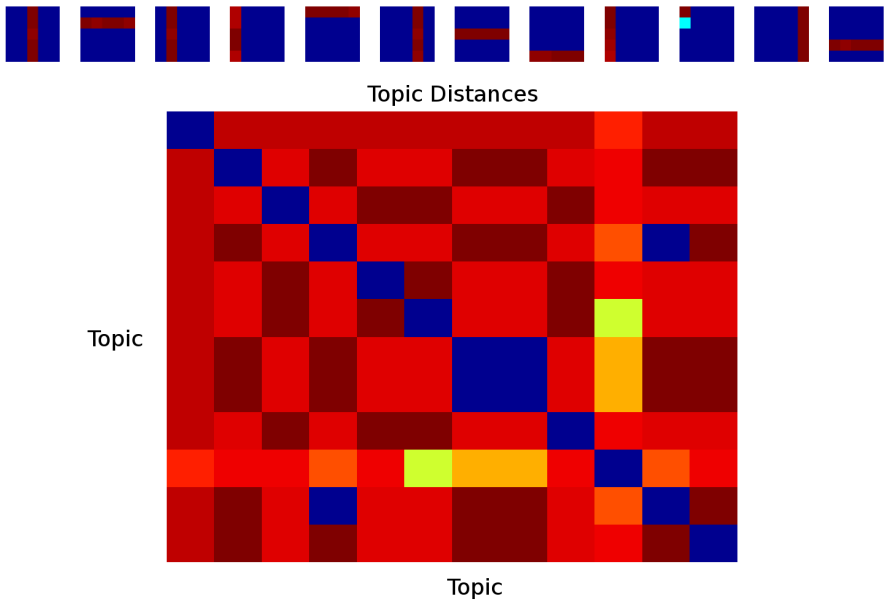


Figure 4.15: Topics recovered and the distance between each pair expressed as a matrix when fitting 13 topics.

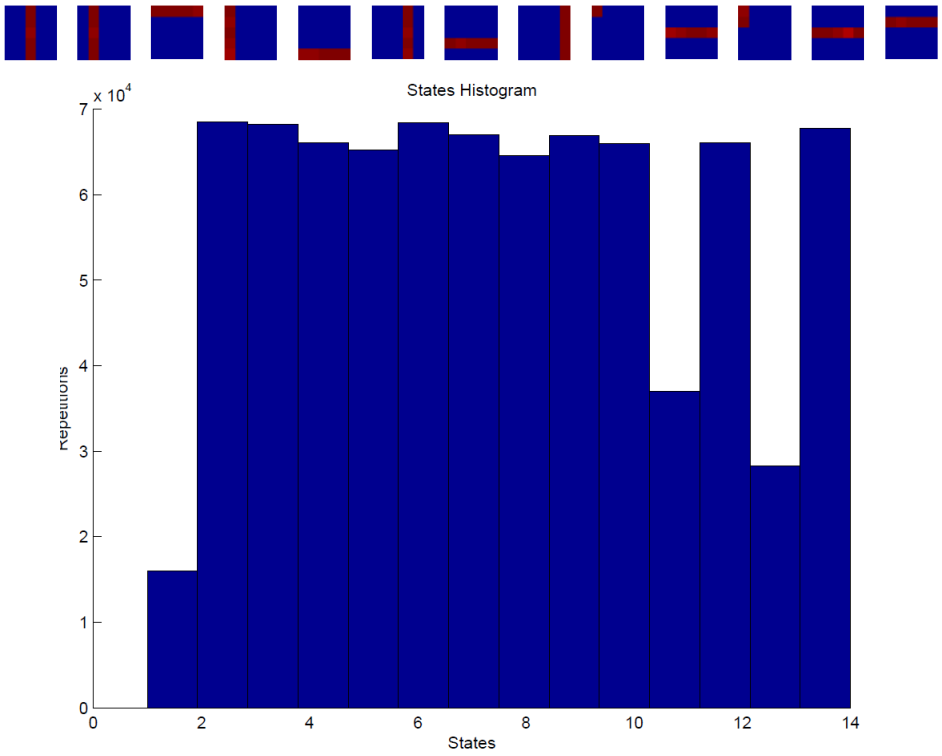


Figure 4.16: Topics recovered by trying to fit 14 together with a histogram of their appearance.

4.5 Synthetic experiment 4

In the experiment in section 4.4 the repetitions from a class were not related one to another, so now we will create a “base sample" and the repetitions will be alterations of it. In the base sample every state z appears just in order and has a random duration z_d, for the repetitions the states appear in the same order, but their duration will be given by a uniform distribution $z_d^r \sim U(z_d - \delta z_d, z_d + \delta z_d)$ where $\delta = .2, .4, .5, .8, 1$. The reason to do this is that the data we intend to analyse with this model appears to have some shift in time and have variable duration for every state.

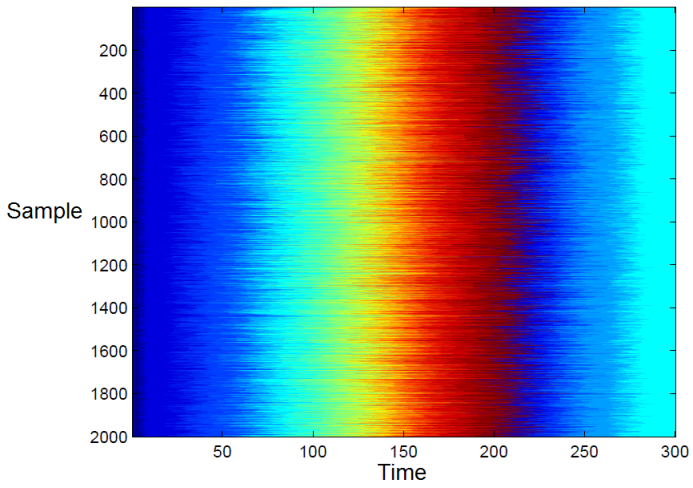


Figure 4.17: States for $\delta = .2$.

Once the states are determined, the topics from 4.4 are used to generate the data.

Using the tools mentioned before the correct number of topics is recovered and the tags are compared with the original ones.

4.5.1 Discussion

A near perfect match is achieved for the topics. The miss-matched tags are mostly associated with topics that are not correctly recovered due to poor

δ	%correct tags
.2	. 96
.4	.90
.5	.88
.8	.93
1	.89

Figure 4.18: Effectiveness of the state recovery for different elongations in time.

representation of them in the original data as for example with the first state in figure 4.17.

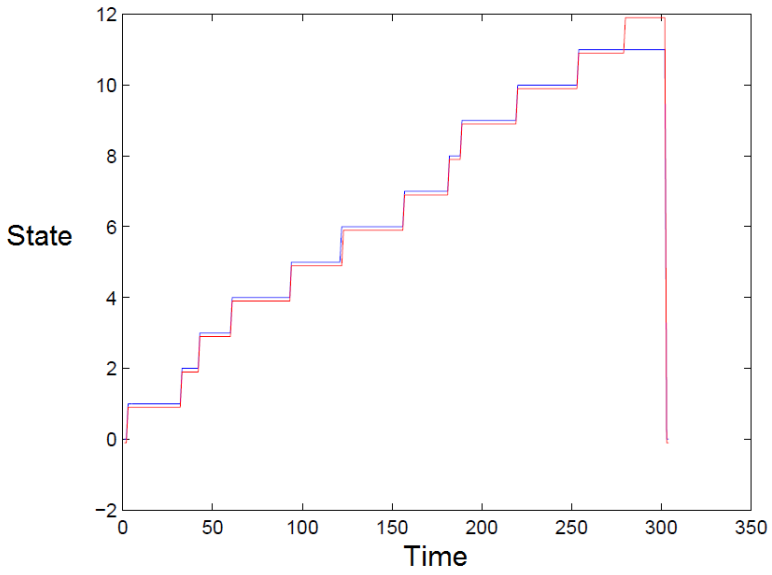


Figure 4.19: Matching of the recovered labels to the real ones.

Figure 4.20: Topics recovered with $\delta = .2$. It can be seen that the original topics 1, 11 and 12 are mixed into state 6 (from left to right in the image) since they all appear too few times in the data and are similar.

Since we usually don't have the real tags when analysing the data, two

useful visualizations for the data are used. First, for a given time, the most likely state over all the repetitions on that class is calculated and graphed as in figure 4.21. The problem is that a state with low duration may be shadowed by bigger ones.

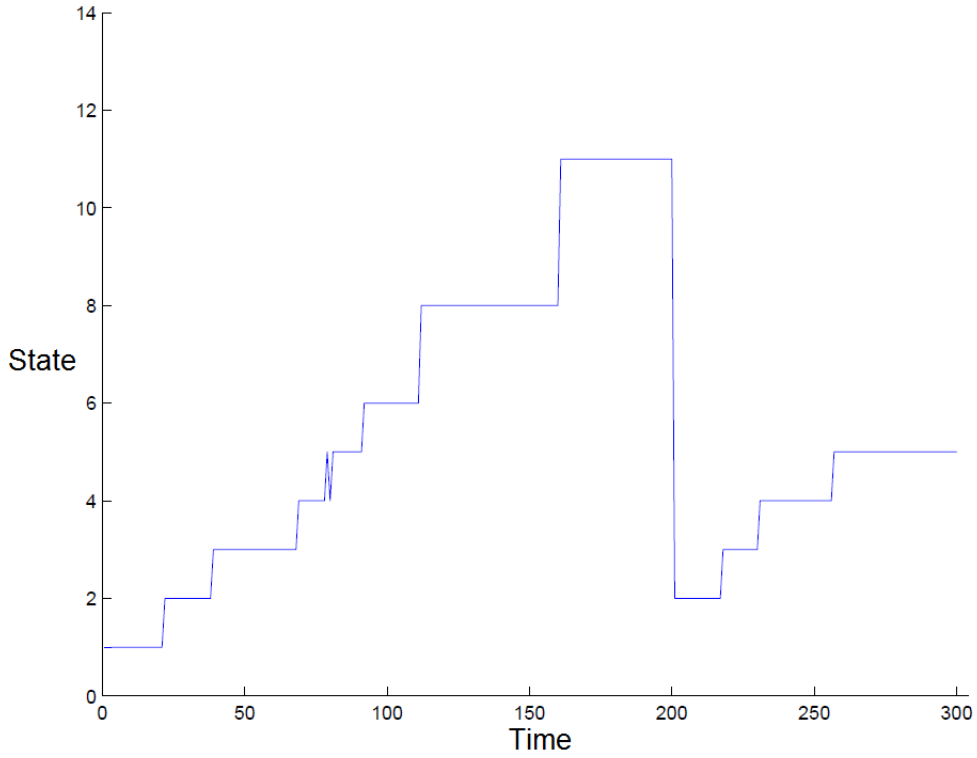


Figure 4.21: The most likely state at each time is calculated and the labels are ordered to allow an easier visualization.

The second is to graph the empirical probability of all the states for each time and graph it as in figure 4.22. This visualization will show all the states even if they have very low duration, but just because of that, the curves may overlap too much and difficult the analysis of the graph. Though that overlapping may give some insight on the behaviour of the data.

The algorithm theoretically behaves well with data that is shifted or elongated through time and the experiments support it, so if all the repetitions follow the same overall structure, we are able to recover it even though the state duration varies from one to another. The problem comes with the vi-

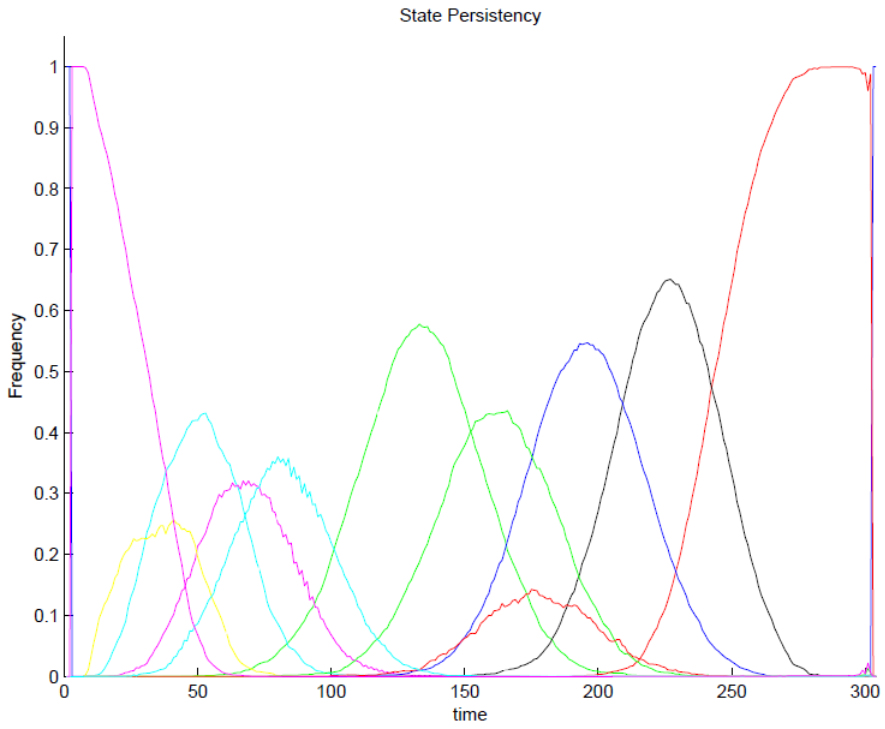


Figure 4.22: The empirical probability of all the states over the time averaging over the repetitions.

sualizations, since there is no obvious way to represent the recovered tags when there is more than one repetition for a experiment and for example the visualization on figure 4.21 will lose states if the elongation is too big. This is very important as our final application is analysing EEG data which suffers from this elongation.

Other issue we want to address with the real data is to distinguish if two a priori different classes are actually different. For this we propose using individual histograms for the state frequency, usually they are different enough to separate the classes as in figure 4.23.

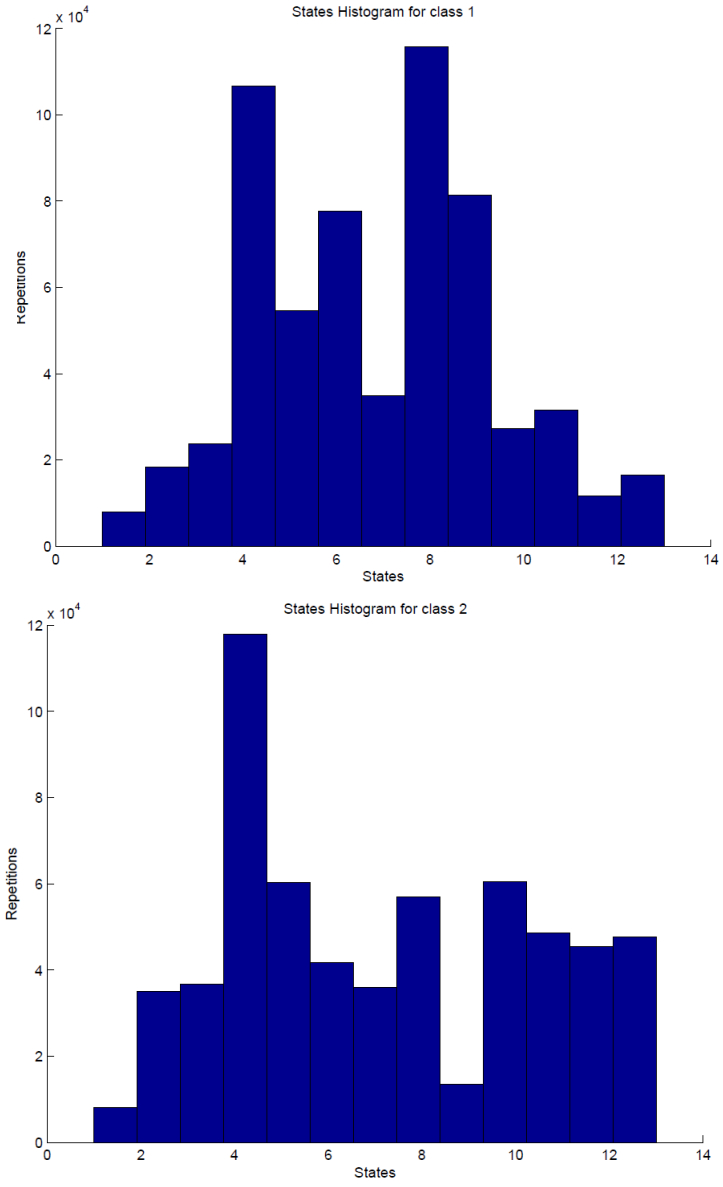


Figure 4.23: Histograms for the different classes on experiment $\delta = .4$. State 9 is evidently important for class 1 and has very little appearance on class 2, helping us differentiate them.

4.6 EEG sample data

We now run the model for a real EEG signal data and analyse it with all the tools we have described. The data we use comes from two experiments, in the first the stimulus is a word representing an animal and in the second a word representing an object, we have 1600 samples from each provided by the work in [9]. We code the data as described in section 1.5, the discretized time is divided in 200 intervals, we focus on just one electrode and the frequencies from 4Hz to 8Hz. This combination of frequencies and electrodes are picked due to the previous analysis done in [9], the discarding of the deactivation information is done due to our belief of it not being as relevant and doing so greatly improves the run time and eases the analysis. We start by looking at the topic versus likelihood graphic in figure 4.24. It can be seen that after 20 topics the behaviour is very erratic, but still the maximum is obtained with 50 topics.

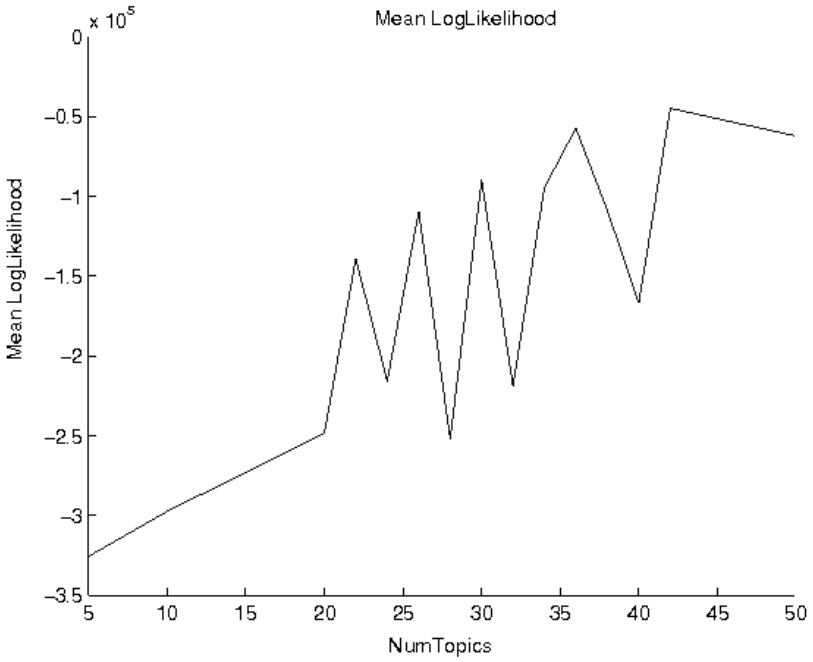


Figure 4.24: Mean log likelihood for the sampled topics: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 .

,

By looking closely at the topics recovered when 50 topics are fitted (Fig. 4.25), we notice that several of them are repeated. So we turn to the topic distance matrix (Fig. 4.26) and see that indeed several of them are actually the same topic.

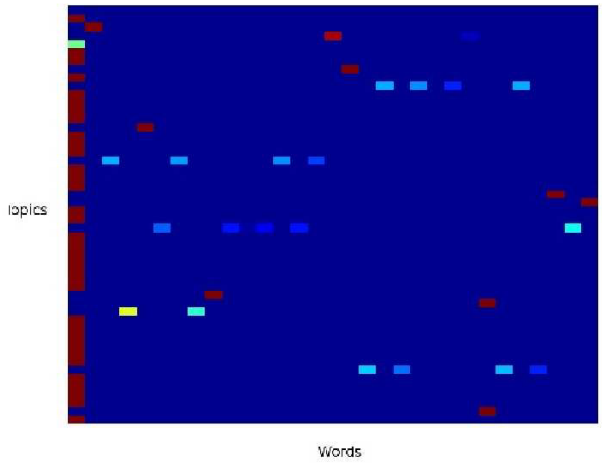


Figure 4.25: Topics recovered when trying to fit 50.

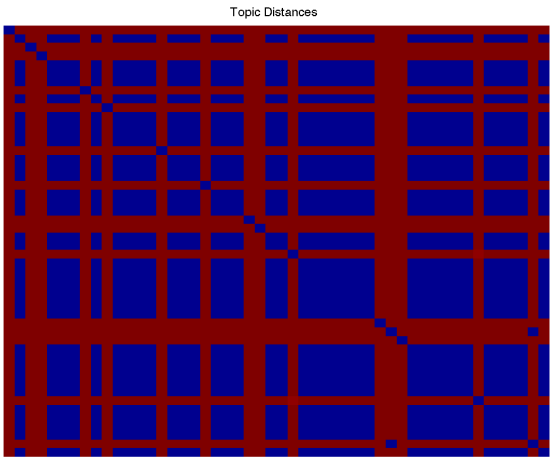


Figure 4.26: Topic distance matrix when trying to fit 50.

We proceed to merge all these topics into one (Fig. 4.27) and continue analysing the data. Comparing the histograms of topic apparition among the classes isn't very informative because the first topic overshadows the rest of them as can be seen on figure 4.28. In figure 4.29 we can see a zoom of the state histogram without state 1 and differences between both classes becomes more apparent.

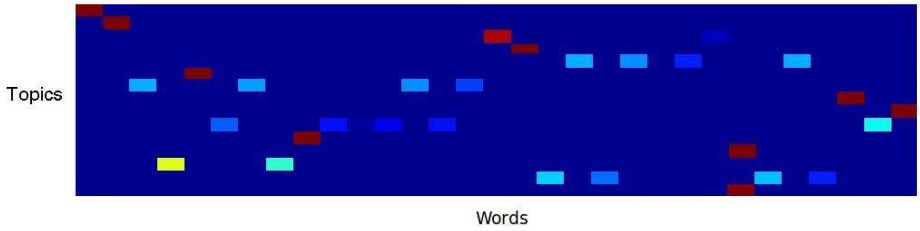


Figure 4.27: Non-repeated topics when trying to fit 50, showing that there are actually just 14.

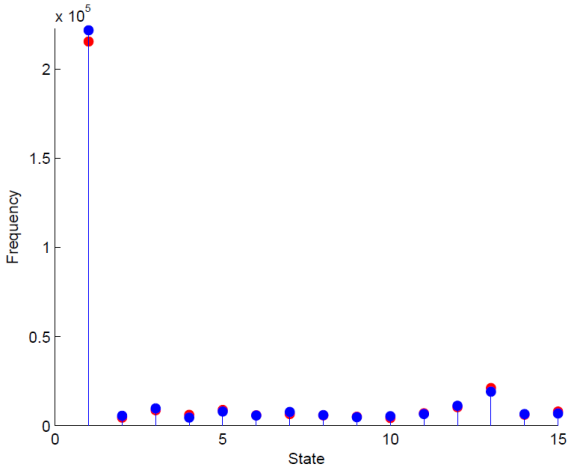


Figure 4.28: Frequency of appearance of each topic for both classes. Class 1 is blue and class 2 red.

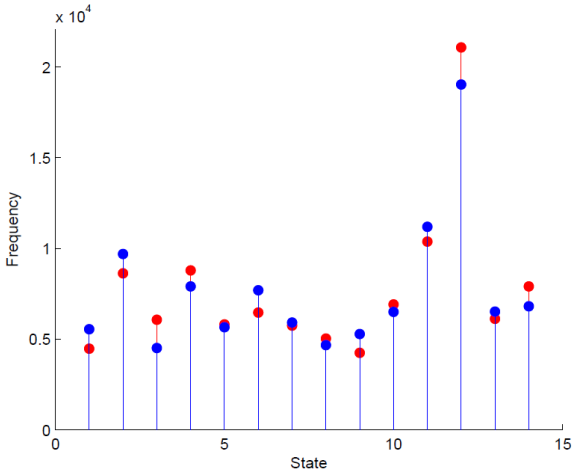


Figure 4.29: Frequency of appearance of each topic except topic 1 for both classes. Class 1 is blue and class 2 red.

When we try looking at the empirical probability of every state over the time (Fig. 4.30), the same difficulty as with the histograms arise: the state 1 overshadows the rest of them. So just like with the histograms, we zoom the graph to ignore the state 1 (Fig. 4.31) and again the differences between both classes becomes more apparent.

4.6.1 Discussion

We hoped to find the correct number of topics just with the help of the likelihood, but this was not possible. Still the findings are encouraging since the other tools (the topic distance matrix) indicate the correct number of topics is 14. It is worth noting that the 14 topics being the correct number isn't entirely unsupported by the likelihood graph, because after going past 15 topics, its behaviour starts being erratic, which hints that maybe the maximum has already passed. This is a very good segmentation in the sense that the data is compressed from a dictionary of 32 words to one of 14.

Throughout the analysis the state 1 has been frequently causing troubles: it was the state that was split into several copies and then overshadowed all the other topics. When we look closely at the topic corresponding to this state, we see that the vast majority of its probability mass is focused on the

word 1, which corresponds to the EEG signal is “dormant" (i.e. there is no activity). This dormant state of the EEG signal is actually where the signal is most of the time, so not only it is not surprising that a full topic is spent solely on it, but it is actually desirable. Also we can see that no other topic assigns probability mass to the word 1, which means that the model is perceiving correctly when there is activity. One other thing we can appreciate from the topics is that all of them distribute their probability mass among 5 words at most, so they are quite sparse. Further analysis has to be performed in order to check if the segmentation actually makes sense physiologically-wise i.e. if the words assigned to the same topic actually represent similar processes in the EEG signal.

Ideally we would want to see a graph showing the state with maximum probability at each time as with synthetic experiment 4. Unfortunately, the state 1 overshadows all the others by far and this graph would only show this state, so it isn't informative.

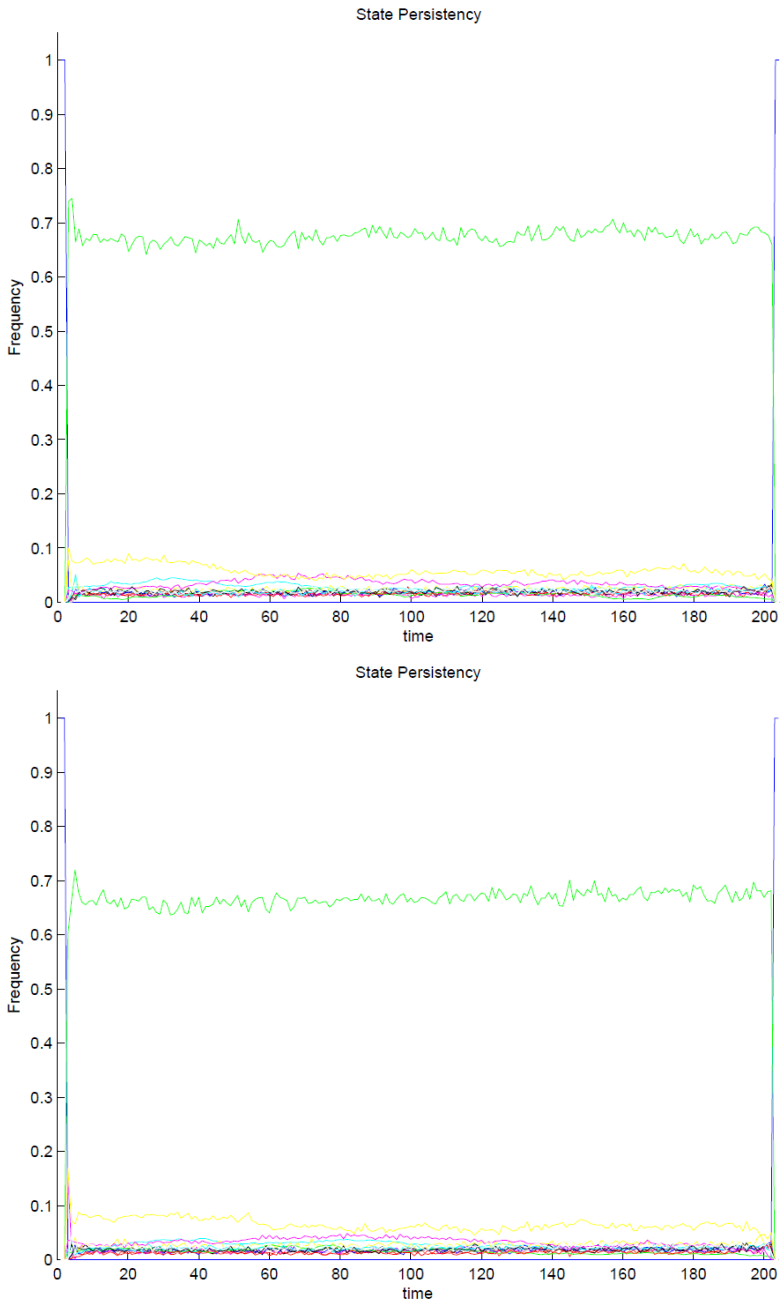


Figure 4.30: Empirical probability of the states over the time averaging over the repetitions.

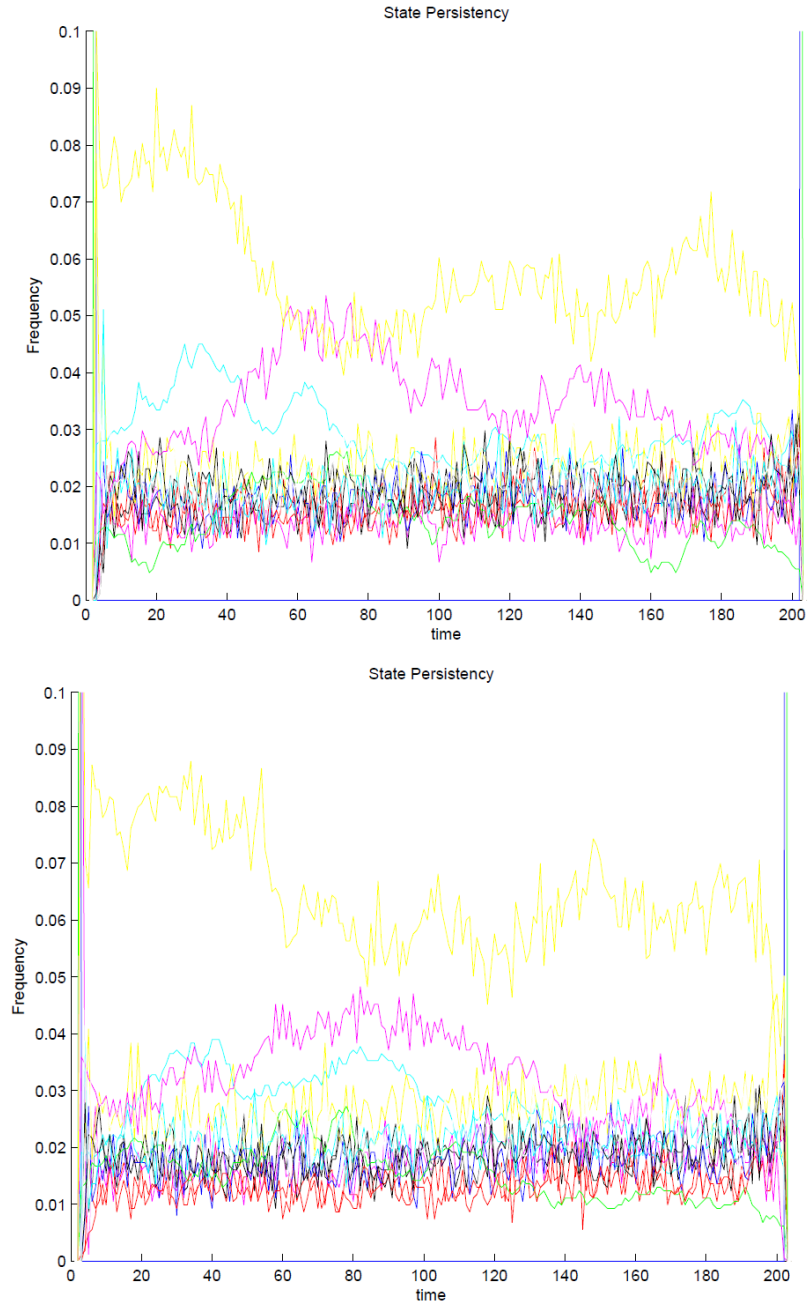


Figure 4.31: Zoom of the empirical probability of the states over the time averaging over the repetitions.

Chapter 5

Conclussion and future work

The models we presented have shown great utility in the data mining process for the EEG signals and semantic analysis. The tools provided greatly help the diagnostic for convergence and posterior analysis of the data after the models are used. As part of the work presented, we began an exploratory analysis of EEG data (shown in section 4.6) that shows great promise and will be continued using the toolbox we developed in CIMAT.

There are several diverse directions in which the study can continue.

- *Evolutionary Markov chain Monte Carlo (EMCMC).* The basic idea behind the EMCMC is to run several chains of the MCMC algorithm and, with a combination of ideas from both Evolutionary Algorithms theory and those of the Metropolis-Hastings algorithm, find a way to mix the chains to improve the overall performance of the chain, i.e. overcome local maxima with more ease.

  Viewed as a genetic algorithm, essentially an iteration of the MCMC algorithm over a chain will be the mutation step and then the crossing of the chains is performed as in a standard genetic algorithm, but after the crossing is done, a Metropolis test is used to accept or reject it. On the other hand, viewed as a MCMC algorithm, the idea is to sample from a multivariate distribution where each of the random variables comes from the desired distribution but the samples are alternately obtained from a standard MCMC algorithm using just the corresponding previous sample and from an "artificial" distribution, that takes two previous samples (instead of just one as in an standard MCMC) and creates a new one.

The advantage of this algorithm is that it has greater exploring capacity than a standard MCMC, but it is far more resource demanding. In particular for our application, we believe it is not feasible due to the large number of variables[14].

- *Reversible Jump Markov chain Monte Carlo (RJMCMC).* As we saw throughout the work, determining the correct number of subjacent states for the data is not an easy problem. The objective of the RJMCMC is to bypass this issue by treating the number of states as a random variable itself and let the MCMC algorithm explore its distribution.

  There are different ways to perform this dimension jump (increasing or decreasing the states), the most popular due to its simplicity and effectiveness is the birth and death process, where a topic is created (birth) using the prior distribution, the inverse process (death) takes one topic at random and "kills it"; after the topics are modified (created or destroyed), the data is reallocated and a Metropolis test is used to decide whether the new sample is accepted or not. Typically for the reallocation, a maximum likelihood algorithm is used (such as the forward-backward or Viterbi algorithm). There are other dimensional jump techniques in the literature, such as merging two topics into one or taking one and splitting it (which are analogue to death and birth respectively).

  The difficult of the RJMCMC is that when the number of states changes, the dimensionality of the other variables may change (for example the transition matrix in the HMM) and it is not easy to deal with, the prior distribution has to be sampled and this change has to be taken into account for the Metropolis test for the acceptance of the overall change of the sample[15]. Also there is no apparent way to integrate out the parameters, which is desirable for greater robustness, implementation simplicity and its lower cost computationally wise.

- *Higher order Autoregressive Hidden Markov Model (HoAHMM).* Just like with the change from LDA to HMM and then from HMM to AHMM, we may keep pushing the time dependencies to a higher order. The problem with this is that in the non-collapsed version, the size of the transition hyper-cube grows exponentially, so it quickly becomes untractable. The collapsed version suffers a similar problem, but in

this case, the equations for the derivation that are already somewhat complex for the second degree version, quickly become untractable as it is a complex combinatorics problem. Nevertheless, it is our belief that it is possible to expand the model at least a couple of times in the order.

- One difficulty we encountered when analysing the EEG data is that the signals we were using were not in phase one to another. That is, even if the same processes occurred on two different samples, they occur at different slightly different moments and at with different durations. This is not an issue per se for the models, as they successfully recover the structure, but for the purposes of further analysis it becomes a big problem. One concrete example we found is: for most of the time, the signal is in a “deactivated" state, and the activation periods are very short in duration, so when we average over the samples, we obtain a signal that is deactivated all of the time even when there are activation periods on all of the individual samples. There is no obvious way as to how to manipulate the signals so they are in phase and averaging them doesn't lose all of the information and it is a subject that needs further study.

Developing this work helped me to understand the general Bayesian paradigm and how some real life problems are being solved by utilizing it. Concretely in the area of data mining, since the models studied are a state of the art technique and have a wide variety of applications. Also the deep study of the LDA and its variations that was necessary for understanding and implementing the HMM and aHMM as well as the EMCMC and RJMCMC (which were studied but out of the scope of this work), made apparent how easy it is to manipulate in order to fit the specific needs of different problems.

Appendix A

Algorithms

In this appendix we present the algorithms mentioned in the text that were not explicitly shown.

For the acceptance function

$$\alpha(X_i^t, Y_i) = \min\left(1, \frac{\pi(Y_i|X_{-i}^t)q_i(X_i^t|Y_i, X_{-i}^t)}{\pi(X_i|X_{-i})q_i(Y_i|X_i, X_{-i})}\right).$$

Algorithm 7 Single-component Metropolis - Hastings

Initialize X^0 at random.
for $t = 1$: Number of Samples **do**
  for $i = 1 : I$ **do**
    Sample $Y_i \sim q_i(\cdot|X_{-i}^t)$.
    Sample $u \sim U(0, 1)$.
    if $u \leq \alpha(X_i^t, Y_i)$ **then**
      set $X_i^{t+1} = Y_i$.
    else
      set $X_i^{t+1} = X_i^t$.
    end if
  end for
end for

Algorithm 8 Single-component Metropolis - Hastings Random-Scan

Initialize X^0 at random.
for $t = 1 : \text{Number of Samples}$ **do**
  Sample $i \sim U(1, ..., I)$.
  Sample $Y_i \sim q_i(\cdot|X^t_{-i})$.
  Sample $u \sim U(0, 1)$.
  if $u \leq \alpha(X^t_i, Y_i)$ **then**
    set $X^{t+1}_i = Y_i$.
  else
    set $X^{t+1}_i = X^t_i$.
  end if
end for

Appendix B

HMM derivation

The prior for the parameters are:

- $P(z_{jr0}|w_j) = \prod_{l=1}^{K} w_{j,l}^{z_{jr0,l}}$.
- $P(z_{jrt}|A_j, z_{jr(t-1)}) = \prod_{k=1}^{K}\prod_{l=1}^{K} A_{j,kl}^{z_{jrt,l}\cdot z_{jr(t-1),k}}$.
- $P(x_{jrt}|z_{jrt}, \varphi) = \prod_{l=1}^{K}\prod_{w=1}^{W} \varphi_{l,w}^{z_{jrt,l}\cdot x_{jrt,w}}$.
- $P(A_j|\alpha) = \prod_{k=1}^{K} \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\prod_{l=1}^{K} A_{jkl}^{\alpha-1}$.
- $P(w_j|\gamma) = \frac{\Gamma(K\gamma)}{\Gamma(\gamma)^K}\prod_{l=1}^{K} w_{j,l}^{\gamma-1}$.
- $P(\varphi|\beta) = \frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\prod_{w=1}^{W} \varphi_{k,w}^{\beta-1}$.

From the graphical model we obtain the following factorization of the joint distribution

$$P(z, x, A, w, \varphi|\alpha, \beta) = \prod_{j=1}^{J} P(A^j, \alpha)P(w^j|\alpha) \tag{B.1}$$

$$\prod_{j=1}^{J}\prod_{r=1}^{R} P(z_{jr0}|w_j)\prod_{t=1}^{T} P(z_{jrt}|z_{jr(t-1)}, A_j) \tag{B.2}$$

$$\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=0}^{T} P(x_{jrt}|z_{jrt}, \varphi) \tag{B.3}$$

$$P(\beta)\prod_{k=1}^{K}P(\varphi_k|\beta) \tag{B.4}$$

B.0.2 Collapsed Model

Expanding B.1

$$\begin{aligned}\prod_{j=1}^{J}P(A_j|\alpha)P(w_j|\gamma) &= \left[\prod_{j=1}^{J}\prod_{k=1}^{K}\frac{\Gamma(K\alpha)}{\Gamma(\alpha)}^{K}\prod_{l=1}^{K}A_{jkl}^{\alpha-1}\right]\left[\prod_{j=1}^{J}\frac{\Gamma(K\gamma)}{\Gamma(\gamma)^{K}}\prod_{l=1}^{K}w_{j,l}^{\gamma-1}\right]\\ &= \left[\frac{\Gamma(K\alpha)^{JK}}{\Gamma(\alpha)^{JK^2}}\prod_{j=1}^{J}\prod_{k=1}^{k}\prod_{l=1}^{K}A_{j,kl}^{\alpha-1}\right]\left[\frac{\Gamma(K\alpha)^{J}}{\Gamma(\alpha)^{JK}}\prod_{j=1}^{J}\prod_{l=1}^{K}w_{j,l}^{\gamma-1}\right]\end{aligned}$$

Expanding B.2

$$\begin{aligned}\prod_{j=1}^{J}\prod_{r=1}^{R}P(z_{jr0}|w_j)\prod_{t=1}^{T}P(z_{jrt}|z_{jr(t-1)},A_j) &= \left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{l=1}^{K}w_{j,l}^{z_{jr0,l}}\right]\\ &\quad\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T}\prod_{k=1}^{K}\prod_{l=1}^{K}A_{j,kl}^{z_{jrt,l}\cdot z_{jr(t-1),k}}\right]\\ &= \left[\prod_{j=1}^{J}\prod_{l=1}^{K}w_{j,l}^{\sum_{r=1}^{R}z_{jr0,l}}\right]\\ &\quad\left[\prod_{j=1}^{J}\prod_{k=1}^{k}\prod_{l=1}^{K}A_{j,kl}^{\sum_{t=1}^{T}\sum_{r=1}^{R}z_{jrt,l}\cdot z_{jr(t-1),k}}\right]\\ &= \left[\prod_{j=1}^{J}\prod_{l=1}^{K}w_{j,l}^{v_{jl}}\right]\left[\prod_{j=1}^{J}\prod_{k=1}^{k}\prod_{l=1}^{K}A_{j,kl}^{\eta_{jkl}}\right]\end{aligned}$$

taking

$$v_{jl}=\sum_{r=1}^{R}z_{jr0,l}$$

and

$$\eta_{jkl}=\sum_{t=1}^{T}\sum_{r=1}^{R}z_{jrt,l}\cdot z_{jr(t-1),k},$$

which represent the number of times state k is selected and the number of times that l is chosen after state k in all samples respectively.

Expanding B.3

$$\begin{aligned}
\prod_{r=1}^{R}\prod_{j=1}^{J}\prod_{t=0}^{T} P(x_{jrt}|z_{jrt},\varphi) &= \prod_{r=1}^{R}\prod_{j=1}^{J}\prod_{t=0}^{T}\prod_{l=1}^{K}\prod_{w=1}^{W}\varphi_{l,w}^{z_{jrt,l}\cdot x_{jrt,w}} \\
&= \prod_{l=1}^{K}\prod_{w=1}^{W}\prod_{r=1}^{R}\prod_{j=1}^{J}\prod_{t=0}^{T}\varphi_{l,w}^{z_{jrt,l}\cdot x_{jrt,w}} \\
&= \prod_{l=1}^{K}\prod_{w=1}^{W}\varphi_{l,w}^{\sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{t=0}^{T} z_{jrt,l}\cdot x_{jrt,w}} \\
&= \prod_{l=1}^{K}\prod_{w=1}^{W}\varphi_{l,w}^{n_{l,w}},
\end{aligned}$$

taking

$$n_{lw} = \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{t=0}^{T} z_{jrt,l}\cdot x_{jrt,w},$$

which is the number of times the state l appears in z together with the state w in x in all samples.

Expanding B.4

$$\prod_{k=1}^{K} P(\varphi_k|\beta) = \frac{\Gamma(W\beta)^K}{\Gamma(\beta)^{WK}}\prod_{k=1}^{K}\prod_{w=1}^{W}\varphi_{k,w}^{\beta-1} \tag{B.5}$$

Substituting on the original formula

$$\begin{aligned}
P(z,x,A,w,\varphi|\alpha,\beta) = &\left[\frac{\Gamma(K\alpha)^{JK}}{\Gamma(\alpha)^{JK^2}}\prod_{j=1}^{J}\prod_{k=1}^{k}\prod_{l=1}^{K}A_{j,kl}^{\alpha-1}\right]\left[\frac{\Gamma(K\alpha)^{J}}{\Gamma(\alpha)^{JK}}\prod_{j=1}^{J}\prod_{l=1}^{K}w_{j,l}^{\gamma-1}\right] \\
&\left[\prod_{j=1}^{J}\prod_{l=1}^{K}w_{j,l}^{v_{jl}}\right]\left[\prod_{j=1}^{J}\prod_{k=1}^{k}\prod_{l=1}^{K}A_{j,kl}^{\eta_{jkl}}\right] \\
&\prod_{l=1}^{K}\prod_{w=1}^{W}\varphi_{l,w}^{n_{l,w}}
\end{aligned}$$

$$
\begin{aligned}
= & \left[\frac{\Gamma(K\alpha)^J}{\Gamma(\alpha)^{JK}}\prod_{j=1}^{J}\prod_{l=1}^{K} w_{jl}^{v_{jl}+\alpha-1}\right] \\
& \left[\frac{\Gamma(K\alpha)^{JK}}{\Gamma(\alpha)^{JK^2}}\prod_{j=1}^{J}\prod_{k=1}^{K}\prod_{l=1}^{K} A_{jkl}^{\eta_{jkl}+\alpha-1}\right] \\
& \left[\frac{\Gamma(W\beta)^K}{\Gamma(\beta)^{KW}}\prod_{k=1}^{K}\prod_{w=1}^{W} \varphi_{kw}^{n_{kw}+\beta-1}\right]
\end{aligned}
$$

and integrating the A^j, w^j and φ we get

$$
\begin{aligned}
P(z,x|\alpha,\beta) = & \frac{\Gamma(K\alpha)^{J(K+1)}}{\Gamma(\alpha)^{KJ}}\frac{\Gamma(K\alpha)^{JK}}{\Gamma(\alpha)^{JK^2}} \\
& \left[\prod_{j=1}^{J}\frac{\prod_{k=1}^{K}\Gamma(v_{jk}+\alpha)}{\Gamma(v_{j\cdot}+K\alpha)}\right] \\
& \left[\prod_{j=1}^{J}\prod_{k=1}^{K}\frac{\prod_{l=1}^{K}\Gamma(n_{jkl}+\alpha)}{\Gamma(n_{jk\cdot}+K\alpha)}\right] \\
& \left[\prod_{k=1}^{K}\frac{\prod_{w=1}^{W}\Gamma(n_{kw}+\beta)}{\Gamma(n_{k\cdot}+W\beta)}\right].
\end{aligned}
$$

Finally, by isolating the terms dependent on z_{jrt} we get

$$
\begin{aligned}
P(z_{jrt,s}=1|-) = & \frac{\nu_{sw}+\beta}{\nu_{s\cdot}+W_s\beta}\frac{\eta_{jks}+\alpha}{\eta_{ck\cdot}+K\alpha} \\
& \frac{\eta_{csr}+I(k=s)I(r=s)+\alpha}{\eta_{cs\cdot}+I(k=s)+K\alpha},
\end{aligned}
$$

where η_{jks} is the number of times in experiment J that state s is obtained after state k; $\eta_{jk\cdot}$ the number of times in experiment J that state k appears; ν_{sw} the times state s appears together with word w and $\nu_{s\cdot}$ the total number of times state s appears.

B.0.3 Non collapsed model

We will derive the full conditional distributions for all the variables.

Notation:
$\mathbf{z} = \{z_{jrt} : j = 1, ..., J \; r = 1, ..., R, \; t = 0, ..., T\}$
$\mathbf{x} = \{x_{jrt} : j = 1, ..., J \; r = 1, ..., R, \; t = 1, ..., T\}$
$\varphi = \{\varphi_k\}_{k=1}^{K}$

$$
\begin{aligned}
a) \;\; P(\alpha^j | A^j, w^j) &\propto P(A^j, w^j, \alpha^j) \\
&\propto P(A^j | \alpha^j) P(w^j | \alpha^j) P(\alpha^j) \\
&\propto \left[\prod_{k=1}^{K} P(A_k^j | \alpha^j) \right] P(w^j | \alpha^j) P(\alpha^j) \\
&\propto U(0,1) \prod_{k=1}^{K} Dir(\alpha^j) Dir(\alpha^j) \\
&\propto U(0,1) \left[\prod_{k=1}^{K} \prod_{l=1}^{K} (A_{kl}^j)^{\alpha - 1} \right] \prod_{l=1}^{K} (w_l^j)^{\alpha - 1}
\end{aligned}
$$

$$
\begin{aligned}
b) \;\; P(w^j | \alpha^j, z_{j10}, ..., z_{jR0}) &\propto P(w^j, \alpha^j, z_{j10}, ..., z_{jR0}) \\
&\propto P(\alpha^j) P(w^j | \alpha^j) \prod_{r=1}^{R} P(z_{jr0} | w^j) \\
&\propto \prod_{r=1}^{R} P(z_{jr0} | w^j) P(w^j | \alpha^j) \\
&\propto \prod_{r=1}^{R} Mult(w^j) Dir(\alpha^j) \\
&\propto \prod_{r=1}^{R} \prod_{l=1}^{K} (w_l^j)^{z_{jr0,l}} \prod_{l=1}^{K} (w_l^j)^{\alpha^j - 1} \\
&\propto \prod_{l=1}^{K} (w_l^j)^{\sum_{r=1}^{R} z_{jr0,l}} \prod_{l=1}^{K} (w_l^j)^{\alpha^j - 1} \\
&\propto \prod_{l=1}^{K} (w_l^j)^{\sum_{r=1}^{R} z_{jr0,l} + \alpha^j - 1} \\
&\propto Dir((\sum_{r=1}^{R} z_{jr0,1}, \sum_{r=1}^{R} z_{jr0,2}, ..., \sum_{r=1}^{R} z_{jr0,K}) + \alpha^j)
\end{aligned}
$$

$$
\begin{aligned}
c)\quad P(A_k^j|\alpha^j,\mathbf{z},A_{-k}^j) &\propto P(A^j,\alpha^j,\mathbf{z})\\
&\propto P(\alpha^j)P(A^j|\alpha^j)P(\mathbf{z}|A^j)\\
&\propto \prod_{l=1}^{K}P(A_l^j|\alpha^j)\left[P(z_{jr0})\prod_{r=1}^{R}\prod_{t=1}^{T}P(z_{jrt}|z_{jr(t-1)},A^j)\right]\\
&\propto P(A_k^j|\alpha^j)\prod_{r=1}^{R}\prod_{t=1,z_{jr(t-1)}=k}^{T}P(z_{jrt}|z_{jr(t-1)},A^j)\\
&\propto Dir(\alpha^j)\prod_{r=1}^{R}\prod_{t=1,z_{jr(t-1)}=k}^{T}Mult(A^j_{z_{jr(t-1)}})\\
&\propto \prod_{l=1}^{K}(A_{kl}^j)^{\alpha^j-1}\prod_{r=1}^{R}\prod_{t=1,z_{jr(t-1)}=k}^{T}\prod_{m=1}^{K}\prod_{l=1}^{K}(A_{ml}^j)^{z_{jrt,l}z_{jr(t-1),m}}\\
&\propto \prod_{l=1}^{K}(A_{kl}^j)^{\alpha^j-1}\prod_{r=1}^{R}\prod_{t=1,z_{jr(t-1)}=k}^{T}\prod_{l=1}^{K}(A_{kl}^j)^{z_{jrt,l}}\\
&\propto \prod_{l=1}^{K}(A_{kl}^j)^{\alpha^j-1}\prod_{l=1}^{K}(A_{kl}^j)^{\sum_{r=1}^{R}\sum_{t=1,z_{jr(t-1)}=k}^{T}z_{jrt,l}}\\
&\propto \prod_{l=1}^{k}(A_{kl}^j)^{\alpha^j-1+\sum_{r=1}^{R}\sum_{t=1,z_{jr(t-1)}=k}^{T}z_{jrt,l}}\\
&\propto Dir\left(\alpha^j+\left(\sum_{r=1}^{R}\sum_{t=1,z_{jr(t-1)}=k}^{T}z_{jrt,l}\right)_{l=1}^{K}\right)
\end{aligned}
$$

$$
\begin{aligned}
d)\quad P(z_{jr0}\ \ |\ \ w^j,x_{jr0},z_{jr1},A^j,\varphi) &\propto P(z_{jr0},w^j,x_{jr0},z_{jr1},A^j,\varphi)\\
&\propto P(A^j)P(w^j)P(z_{jr0}|w^j)P(\varphi)P(x_{jr0}|z_{jr0},\varphi)P(z_{jr1}|z_{jr0},A^j)\\
&\propto P(z_{jr0}|w^j)P(x_{jr0}|z_{jr0},\varphi)P(z_{jr1}|z_{jr0},A^j)\\
&\propto Mult(w^j)Mult(\varphi_{z_{jr0}})Mult(A^j_{z_{jr0}})\\
&\propto \prod_{l=1}^{K}(w_l^j)^{z_{jr0,l}}\prod_{l=1}^{K}\prod_{w=1}^{W}(\varphi_w^l)^{z_{jr0,l}x_{jr0,w}}
\end{aligned}
$$

$$\prod_{k=1}^{K}\prod_{l=1}^{K}(A_{kl}^{j})^{z_{jr1,l}z_{jr0,k}}$$

$$\begin{aligned}
e)\ P(z_{jrt} \mid z_{jr(t-1)}, z_{jr(t+1)}, A^j, x_{jrt}, \varphi) &\propto P(z_{jrt}, z_{jr(t-1)}, z_{jr(t+1)}, A^j, x_{jrt}, \varphi) \\
&\propto P(\varphi)P(A^j)P(x_{jrt}|z_{jrt})P(z_{jrt}|z_{jr(t-1)}, A^j) \\
&\quad P(z_{jr(t+1)}|z_{jrt}, A^j) \\
&\propto P(x_{jrt}|z_{jrt})P(z_{jrt}|z_{jr(t-1)}, A^j)P(z_{jr(t+1)}|z_{jrt}, A^j) \\
&\propto Mult(\varphi_{z_{jrt}})Mult(A^j_{z_{jr(t-1)}})Mult(A^j_{z_{jrt}}) \\
&\propto \prod_{l=1}^{K}\prod_{w=1}^{W}(\varphi_w^l)^{z_{jrt,l}x_{jrt,w}}\prod_{k=1}^{K}\prod_{l=1}^{K}(A_{kl}^j)^{z_{jrt,l}z_{jr(t-1),k}} \\
&\quad \prod_{k=1}^{K}\prod_{l=1}^{K}(A_{kl}^j)^{z_{jr(t+1),l}z_{jrt,k}}
\end{aligned}$$

$$\begin{aligned}
f)\ P(z_{jrT}|z_{jr(T-1)}, A^j, x_{jrT}, \varphi) &\propto P(z_{jrT}, z_{jr(T-1)}, A^j, x_{jrT}, \varphi) \\
&\propto P(A^j)P(z_{jr(T-1)}|A^j)P(z_{jrT}|A^j, z_{jr(T-1)}) \\
&\quad P(x_{jrT}|z_{jrT}, \varphi)P(\varphi) \\
&\propto P(z_{jrT}|A^j, z_{jr(T-1)})P(x_{jrT}|z_{jrT}, \varphi) \\
&\propto Mult(A^j_{z_{jr(T-1)}})Mult(\varphi_{z_{jrT}}) \\
&\propto \prod_{k=1}^{K}\prod_{l=1}^{K}(A_{kl}^j)^{z_{jrT,l}z_{jr(T-1),k}}\prod_{l=1}^{K}\prod_{w=1}^{W}(\varphi_w^l)^{z_{jrT,l}x_{jrT,w}}
\end{aligned}$$

$$\begin{aligned}
g)\ P(\varphi^k \mid \varphi^{-k}, \beta, \mathbf{x}, \mathbf{z}) &\propto P(\varphi, \beta, \mathbf{x}, \mathbf{z}) \\
&\propto P(\beta)P(\varphi|\beta)P(\mathbf{z})P(\mathbf{x}|\mathbf{z}, \varphi) \\
&\propto P(\varphi|\beta)\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T}P(x_{jrt}|z_{jrt}, \varphi)\right] \\
&\propto P(\varphi|\beta)\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1,z_{jrt}=k}^{T}P(x_{jrt}|z_{jrt}, \varphi)\right]
\end{aligned}$$

$$
\begin{aligned}
&\propto Dir(\beta)\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T} Mult(\varphi^{z_{jrt}})\right] \\
&\propto \prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1,z_{jrt}=k}^{T}\prod_{l=1}^{K}\prod_{w=1}^{W}(\varphi_w^l)^{z_j rt,l x_{jrt,w}}\right] \\
&\propto \prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1,z_{jrt}=k}^{T}\prod_{w=1}^{W}(\varphi_w^k)^{z_j rt,k x_{jrt,w}}\right] \\
&\propto \prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1,z_{jrt}=k}^{T}\prod_{w=1}^{W}(\varphi_w^k)^{x_{jrt,w}}\right] \\
&\propto \prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}\prod_{w=1}^{W}(\varphi_w^k)^{\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k}^{T} x_{jrt,w}} \\
&\propto \prod_{w=1}^{W}(\varphi_w^k)^{\beta-1+\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k}^{T} x_{jrt,w}} \\
&\propto Dir(\beta+(\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k}^{T} x_{jrt,1}, \\
&\qquad \sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k}^{T} x_{jrt,2}, \\
&\qquad \ldots, \sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k}^{T} x_{jrt,1}))
\end{aligned}
$$

$$
\begin{aligned}
h)\ P(\beta|\varphi) &\propto P(\beta,\varphi) \\
&\propto P(\beta)P(\varphi|\beta) \\
&\propto P(\beta)\prod_{k=1}^{K} P(\varphi|\beta) \\
&\propto U(0,1)\prod_{k=1}^{K} Dir(\beta) \\
&\propto U(0,1)\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}
\end{aligned}
$$

Appendix C

Autoregressive model derivation

The prior for the parameters are:

- $P(z_{jrt}|A_j, z_{jr(t-1)}, z_{jr(t-2)}) = \prod_{m=1}^{K}(A^j_{z_{jr(t-2)},z_{jr(t-1)}})^{z_{jrt,m}}$.
- $P(x_{jrt}|z_{jrt}, \varphi) = \prod_{w=1}^{W}(\varphi_w^{z_{jrt}})^{x_{jrt,w}}$.
- $P(A^j_{kl}|\alpha^j) \propto \prod_{m=1}^{K}(A^j_{klm})^{\alpha^j-1}$.
- $P(\varphi^k|\beta) \propto \prod_{w=1}^{W}(\varphi^k_w)^{\beta-1}$.

C.0.4 Collapsed Model

We will manipulate the distribution so we can integrate out parameters.

$$
\begin{aligned}
& P(\mathbf{z}, \mathbf{x}, \mathbf{A}, \alpha, \beta, \varphi) \\
= \; & P(\beta)\left[\prod_{k=1}^{K} P(\varphi^k|\beta)\right]\prod_{j=1}^{J} P(\alpha^j) \\
& \prod_{j=1}^{J} P(A^j|\alpha^j)\prod_{r=1}^{R}\prod_{t=1}^{T} P(z_{jrt}|z_{jr(t-2)}, z_{jr(t-1)}, A^j)P(x_{jrt}|z_{jrt}, \varphi) \\
\propto \; & P(\beta)\left[\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi^k_w)^{\beta-1}\right]\prod_{j=1}^{J} P(\alpha^j) \\
& \prod_{j=1}^{J}\prod_{k_1=1}^{K}\prod_{k_2=1}^{K}\prod_{m=1}^{K}(A^j_{k_2k_1,m})^{\alpha-1}\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T}\prod_{m=1}^{K}(A^j_{z_{jr(t-2)}z_{jr(t-1)},m})^{z_{jrt,m}}
\end{aligned}
$$

$$
\begin{aligned}
&\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T}\prod_{w=1}^{W}(\varphi_w^{z_{jrt}})^{x_{jrt,w}} \\
\propto\ & P(\beta)\left[\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}\right]\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T}\prod_{w=1}^{W}(\varphi_w^{z_{jrt}})^{x_{jrt,w}}\right]\prod_{j=1}^{J}P(\alpha^j) \\
&\left[\prod_{j=1}^{J}\prod_{k_1=1}^{K}\prod_{k_2=1}^{K}\prod_{m=1}^{K}(A^j_{k_2k_1,m})^{\alpha-1}\right]\left[\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T}\prod_{m=1}^{K}(A^j_{z_{jr(t-2)}z_{jr(t-1)},m})^{z_{jrt,m}}\right] \\
\propto\ & P(\beta)\left[\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}\right]\left[\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi_w^k)^{\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1}^{T}x_{jrt,w}I(z_{jrt}=k)}\right] \\
&\prod_{j=1}^{J}P(\alpha^j)\left[\prod_{j=1}^{J}\prod_{k_1=1}^{K}\prod_{k_2=1}^{K}\prod_{m=1}^{K}(A^j_{k_2k_1,m})^{\alpha-1}\right] \\
&\prod_{j=1}^{J}\prod_{k_1=1}^{K}\prod_{k_2=1}^{K}\prod_{m=1}^{K}(A^j_{k_2k_1,m})^{\sum_{r=1}^{R}\sum_{t=1}^{T}z_{jrt,m}I(z_{jr(t-2)}=k2)I(z_{jr(t-1)}=k1)} \\
\propto\ & P(\beta)\left[\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi_w^k)^{\beta-1+\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1}^{T}x_{jrt,w}I(z_{jrt}=k)}\right]\prod_{j=1}^{J}P(\alpha^j) \\
&\prod_{j=1}^{J}\prod_{k_1=1}^{K}\prod_{k_2=1}^{K}\prod_{m=1}^{K}(A^j_{k_2k_1,m})^{\alpha-1+\sum_{r=1}^{R}\sum_{t=1}^{T}z_{jrt,m}I(z_{jr(t-2)}=k2)I(z_{jr(t-1)}=k1)}.
\end{aligned}
$$

Integrating parameters out, fixing $\alpha^j,\ \beta$ and setting

$$v_{k,w} = \sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1}^{T}x_{jrt,w}I(z_{jrt}=k),$$

$$v_{k,\cdot} = \sum_{w=1}^{W}v_{k,w},$$

$$s^j_{k_2,k_1,m} = \sum_{r=1}^{R}\sum_{t=1}^{T}z_{jrt,m}I(z_{jr(t-2)}=k_2)I(z_{jr(t-1)}=k_1),$$

$$s^j_{k_2,k_1,\cdot} = \sum_{m=1}^{K}s^j_{k_2,k_1,m}.$$

We obtain

$$P(\mathbf{z},\mathbf{x}) \propto \left[\prod_{k=1}^{K}\frac{\prod_{w=1}^{W}\Gamma(v_{k,w}+\beta)}{\Gamma(v_{k,\cdot}+W\beta)}\right]\left[\prod_{j=1}^{J}\prod_{k2=1}^{K}\prod_{k1=1}^{K}\frac{\prod_{m=1}^{K}\Gamma(s^{j}_{k_2,k_1,m}+\alpha)}{\Gamma(s^{j}_{k_2,k_1,\cdot}+K\alpha)}\right]$$

and

$$\begin{aligned}
P(\mathbf{z},\mathbf{x}) &= \left[\prod_{k=1}^{K}\frac{\Gamma(W\beta)}{\Gamma(\beta)^{W}}\frac{\prod_{w=1}^{W}\Gamma(v_{k,w}+\beta)}{\Gamma(v_{k,\cdot}+W\beta)}\right] \\
&\quad \left[\prod_{j=1}^{J}\prod_{k2=1}^{K}\prod_{k1=1}^{K}\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^{K}}\frac{\prod_{m=1}^{K}\Gamma(s^{j}_{k_2,k_1,m}+\alpha)}{\Gamma(s^{j}_{k_2,k_1,\cdot}+K\alpha)}\right] \\
&= \frac{\Gamma(W\beta)^{K}}{\Gamma(\beta)^{KW}}\left[\prod_{k=1}^{K}\frac{\prod_{w=1}^{W}\Gamma(v_{k,w}+\beta)}{\Gamma(v_{k,\cdot}+W\beta)}\right] \\
&\quad \frac{\Gamma(K\alpha)^{JK^2}}{\Gamma(\alpha)^{JK^3}}\left[\prod_{j=1}^{J}\prod_{k2=1}^{K}\prod_{k1=1}^{K}\frac{\prod_{m=1}^{K}\Gamma(s^{j}_{k_2,k_1,m}+\alpha)}{\Gamma(s^{j}_{k_2,k_1,\cdot}+K\alpha)}\right],
\end{aligned}$$

finally, by isolating the parameters that depend on z_{jrt} we get

$$\begin{aligned}
&P(z_{jrt,k}=1|-) = \\
&\frac{\nu_{s,w}+\beta}{\nu_{s\cdot}+W_s\beta}\frac{\eta_{j,p_2,p_1,k}+\alpha}{\eta_{j,p_2,p_1}+K\alpha}\frac{\eta_{j,p_1,k,n_1}+I(p_2=p_1=k=n_1)+\alpha}{\eta_{j,p_1,k}+I(p_2=p_1=k)+K\alpha} \\
&\frac{\eta_{j,k,n_1,n_2}+I(p_2=k=n_2,p_1=n_1)+I(p_1=k=n_1=n_2)+\alpha}{\eta_{j,k,n_1}+I(p_2=k,p_1=n_1)+I(p_1=k=n_1)+K\alpha},
\end{aligned}$$

where $\eta_{j,a,b,c}$ is the number of times in experiment J that state c is obtained after state b after a; $\eta_{jab\cdot}$ the number of times in experiment J that state b appears after a; ν_{sw} the times state s appears together with word w; $\nu_{s\cdot}$ the total number of times state s appears, $p_2=t-2$, $p_1=t-1$, $n_1=t+1$ and $n_2=t+2$.

C.0.5 Non collapsed model

We will derive the full conditional distributions for the variables.

$$
\begin{aligned}
a)\ \ P(\alpha^j|A^j) &\propto P(\alpha^j)P(A^j|\alpha^j) \\
&\propto U(0,1)\prod_{k=1}^{K}\prod_{l=1}^{K} P(A^j_{kl}|\alpha^j) \\
&\propto U(0,1)\prod_{k=1}^{K}\prod_{l=1}^{K} Dir(\alpha^j) \\
&\propto U(0,1)\prod_{k=1}^{K}\prod_{l=1}^{K}\prod_{m=1}^{K} (A^j_{klm})^{\alpha^j-1}
\end{aligned}
$$

$$
\begin{aligned}
b)\qquad & P(A^j_{k_0l_0}|\alpha^j,\mathbf{z},A^j_{-k_0l_0}) \propto P(A^j,\alpha^j,\mathbf{z}) \\
\propto\ & P(\alpha^j)\prod_{k=1}^{K}\prod_{l=1}^{K} P(A^j_{kl}|\alpha^j)\prod_{r=1}^{r}\prod_{t=1}^{T} P(z_jrt|A^j,z_{jr(t-2)},z_{jr(t-1)}) \\
\propto\ & P(A^j_{k_0l_0}|\alpha^j)\prod_{r=1}^{r}\prod_{t=1,\ z_{jr(t-2)}=k_0,\ z_{jr(t-1)}=l_0}^{T} P(z_{jrt}|A^j_{k_0l_0},z_{jr(t-2)},z_{jr(t-1)}) \\
\propto\ & Dir(\alpha^j)\prod_{r=1}^{r}\prod_{t=1,\ z_{jr(t-2)}=k_0,\ z_{jr(t-1)}=l_0}^{T} Mult(A^j_{k_0l_0}) \\
\propto\ & \prod_{m=1}^{K}(A^j_{k_0l_0m})^{\alpha^j-1}\prod_{r=1}^{r}\prod_{[t=1,\ z_{jr(t-2)}=k_0,\ z_{jr(t-1)}=l_0]}^{T}\prod_{m=1}^{K}(A^j_{k_0l_0m})^{z_{jrt,m}} \\
\propto\ & \prod_{m=1}^{K}(A^j_{k_0l_0m})^{\alpha^j-1}\prod_{m=1}^{K}(A^j_{k_0l_0m})^{\sum_{r=1}^{r}\sum_{t=1}^{T}\left(z_{jrt,m}I(z_{jr(t-2)}=k_0,\ z_{jr(t-1)}=l_0)\right)} \\
\propto\ & \prod_{m=1}^{K}(A^j_{k_0l_0m})^{\alpha^j+\sum_{r=1}^{r}\sum_{t=1}^{T}\left(z_{jrt,m}I(z_{jr(t-2)}=k_0,\ z_{jr(t-1)}=l_0)\right)-1} \\
=\ & Dir\left(\alpha^j+\left(\sum_{r=1}^{r}\sum_{t=1}^{T}\left(z_{jrt,m}I(z_{jr(t-2)}=k_0,\ z_{jr(t-1)}=l_0)\right)\right)_{m=1}^{K}\right)
\end{aligned}
$$

$$
c)\qquad P(z_{jrt}|A^j,z_{jr(t-2)},z_{jr(t-1)},x_{jrt},z_{jr(t+1)},z_{jr(t+2)},A^j,\varphi)
$$

$$
\begin{aligned}
&\propto P(z_{jr(t-2)},...,z_{jr(t+2)},A^j,x_{jrt},\varphi)\\
&\propto P(A^j)P(z_{jr(t-2)}|A^j)P(z_{jr(t-1)}|A^j,z_{jr(t-2)}P(z_{jrt}|A^j,z_{jr(t-1)},z_{jr(t-2)})\\
&\quad P(z_{jr(t+1)}|A^j,z_{jrt},z_{jr(t-1)})P(z_{jr(t+2)}|A^j,z_{jr(t+1)},z_{jrt})P(\varphi)P(x_{jrt}|\varphi,z_{jrt})\\
&\propto P(z_{jrt}|A^j,z_{jr(t-1)},z_{jr(t-2)})P(z_{jr(t+1)}|A^j,z_{jrt},z_{jr(t-1)})\\
&\quad P(z_{jr(t+2)}|A^j,z_{jr(t+1)},z_{jrt})P(x_{jrt}|\varphi,z_{jrt})\\
&\propto Mult(A^j_{z_{jr(t-2)},z_{jr(t-1)}})Mult(A^j_{z_{jr(t-1)},z_{jrt}})\\
&\quad Mult(A^j_{z_{jrt},z_{jr(t+1)}})Mult(\varphi^{z_{jrt}})\\
&\propto \prod_{m=1}^{K}(A^j_{z_{jr(t-2)},z_{jr(t-1)},m})^{z_{jrt,m}}\prod_{m=1}^{K}(A^j_{z_{jr(t-1)},z_{jrt},m})^{z_{jr(t+1),m}}\\
&\quad \prod_{m=1}^{K}(A^j_{z_{jrt},z_{jr(t+1)},m})^{z_{jr(t+2),m}}\prod(\varphi_w^{z_{jrt}})^{x_{jrt,w}})
\end{aligned}
$$

so,

$$
\begin{aligned}
&P(z_{jrt}=k|A^j,z_{jr(t-2)}=p_2,z_{jr(t-1)}=p_1,x_{jrt}=y,z_{jr(t+1)}=n_1,\\
&\qquad z_{jr(t+2)}=n_2,A^j,\varphi)\\
&\propto A^j_{p_2,p_1,k}A^j_{p_1,k,n_1}A^j_{k,n_1,n_2}\varphi^k_y
\end{aligned}
$$

$$
\begin{aligned}
d)\quad P(\varphi^{k_0}|\varphi^{-k_0},\beta,\mathbf{x},\mathbf{z}) &\propto P(\varphi,\beta,\mathbf{x},\mathbf{z})\\
&\propto P(\beta)P(\mathbf{z})P(\varphi|\beta)P(\mathbf{x}|\mathbf{z},\varphi)\\
&\propto \prod_{k=1}^{k}P(\varphi^k|\beta)\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1}^{T}P(x_{jrt}|z_{jrt},\varphi)\\
&\propto P(\varphi^{k_0}|\beta)\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1,z_{jrt}=k_0}^{T}P(x_{jrt}|z_{jrt},\varphi)\\
&\propto Dir(\beta)\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{t=1,z_{jrt}=k_0}^{T}Mult(\varphi^{k_0})\\
&\propto \prod_{w=1}^{W}(\varphi_w^{k_0})^{\beta-1}\prod_{j=1}^{J}\prod_{r=1}^{R}\prod_{[t=1,z_{jrt}=k_0]}^{T}\prod_{w=1}^{W}(\varphi_w^{k_0})^{x_{jrt,w}}\\
&\propto \prod_{w=1}^{W}(\varphi_w^{k_0})^{\beta-1}\prod_{w=1}^{W}(\varphi_w^{k_0})^{\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k_0}^{T}x_{jrt,w}}
\end{aligned}
$$

$$\propto \prod_{w=1}^{W} (\varphi_w^{k_0})^{\beta-1+\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{t=1,z_{jrt}=k_0}^{T} x_{jrt,w}}$$

$$\propto Dir\left(\beta + \left(\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{[t=1,z_{jrt}=k_0]}^{T} x_{jrt,m}\right)_{w=1}^{W}\right)$$

$e)$ $P(\beta|\varphi) \propto P(\varphi,\beta)$

$$\propto P(\beta)\prod_{k=1}^{K} P(\varphi^k|\beta)$$

$$\propto U(0,1)\prod_{k=1}^{K} Dir(\beta)$$

$$\propto U(0,1)\prod_{k=1}^{K}\prod_{w=1}^{W}(\varphi_w^k)^{\beta-1}$$

Appendix D

Program guide

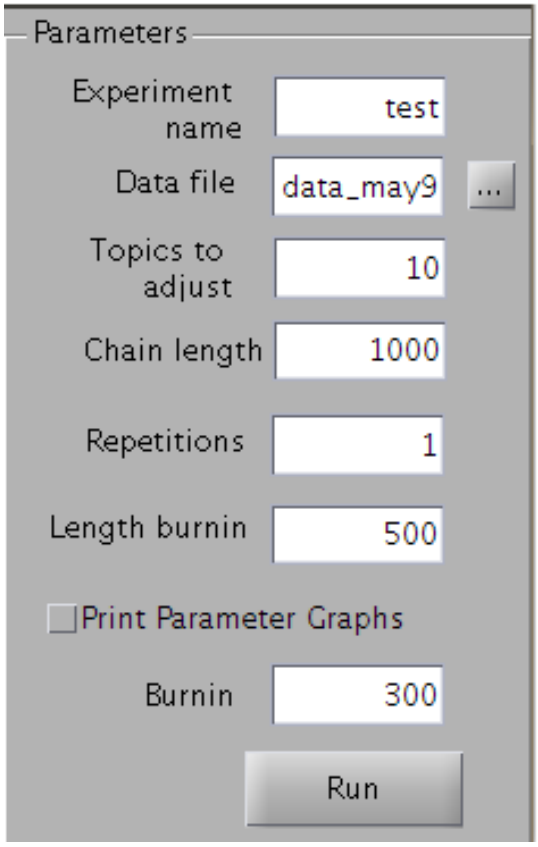


Figure D.1: GUI for the software developed.

- *Experiment name* will be the name of the folder in which all the results will be stored.
- *Data file* is the file containing all the raw data we wish to analise. The data file must contain:
  - C a variable with the number of documents
  - R a variable with the maximum number of repetitions (sentences) on all the documents.
  - *Rspec* a vector with the specific number of repetitions in each document.
  - T a variable with the length of the sentences.
  - W a variable with the total number of words in the dictionary.
  - y a vector of dimension TxRxC containing all the data.
- *Topics to adjust* has all the topics we wish to fit into the data presented as a string, for example "3 5 7 10". The topics can't be repeated.
- *Chain length* is the total number of samples that will be extracted.
- *Repetitions* is the number of times the complete experiment will be repeated. If more than 1 repetition is requested, they will be placed in separate folders named <Experiment name>+<repetition number>.
- *Length burnin* is the length of the chain that will be used for the burnin calculation algorithm.
- *Print Parameter Graphs* will be on if we wish all of the individual graphs to be printed. It will raise the running time, so if we don't need them it is better left off.
- *Burnin* will be the number of samples discarded in case the burnin algorithm determines the MCMC algorithm hasn't converged.
- *Run* after all the parameters are set, this will start running the algorithm.

Bibliography

[1] A. Jasra, C. C. Holmes, D. A. Stephens. Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling. *Statistical Sciennce*. **Vol. 20, No. 1** (2005), 50-67.

[2] C. E. Rasmussen, Z. Ghahramani. Occam's Razor.

[3] C. J. Geyer. Practical Markov Chain Monte Carlo. *Statistical Sciences*. **Vol. 7, No. 4** (1992), 473-511.

[4] D. Aldous. Exchangeability and related topics. In École d'été de probbilités de Saint-Flur, XIII âĂŞ 193, p. 1-198. Springer, Berlin, 1985.

[5] D. M. Blei, A. Y. Ng, M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research* **3** (2003), 993-1022.

[6] D. Novák, T. Al-ani, A. Hamam, L. Lhotská. Electroencephalogram processing using Hidden Markov Models. Czech Technical University in Prague.

[7] G. L. Yang, C. Le, M. Lucien. *Asymptotics in Statistics: Some Basic Concepts*, Springer, Berlin, 2000.

[8] G. N. Avecilla, S. Ruiz, J. L. Marroquin, T. Harmony, A. Alba, O. Mendoza. Electrophysiological auditory responses and language development in infants with preventricular leukomalacia. *Brain and Language*, 2011.

[9] H. G. García. Caracterización electroencefalográfica del procesamiento de la información verbal de una tarea de reconocimiento de palabras. UNAM, Queretaron, 2010.

[10] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer Series in Statistics, New York, 2001.

[11] K. Lee, J. Marin, K. Mengersen, C. Robert. Bayesian Inference on Mixtures of Distributions. (2008)

[12] K. P. Burnham, D.R. Anderson. *Model selection and multimodel inference: A practical Information-Theoretic Approach*, Springer Verlag, New York, 2002.

[13] M. Johnson, T. L. Griffiths, S. Goldwater. Bayesian Inferences for PCFGs via Markov chain Monte Carlo.

[14] M. J. A. Strens. Evolutionary MCMC Sampling and Optimization in Discrete Spaces. *Proceedings of the Twentieth International Conference on Machine Learning* (ICML-2003), Washington DC, 2003

[15] O. Cappé, C. P. Robert, T. Rydén. Reversible jump, birth-and-death, and more general continuous time MCMC samplers.

[16] P. G. Hoel, S. C. Port, C. J. Stone. *Introduction to Probability Theory*, Houghton Mifflin Company, Boston, 1971.

[17] R. J. Boys, D. A. Henderson, A comparison of reversible jump MCMC algorithms for DNA sequence segmentations using hidden Markov models. Department of Statistics, University of Newcastle, U.K.. (2000??)

[18] R. J. Boys, D. A. Henderson, D. J. Wilkinson. Detecting homogeneous segments in DNA sequences by using hidden Markov models. *Applied Statistics*. **49** (2000), 269-285.

[19] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6** (1984), 721-741

[20] S. Goldwater, T. L. Griffiths. A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging.

[21] W.R. Gilkins, S. Richardson, D. J. Spiegelhalter. *Markov Chain Monte Carlo in practice*, Chapman and Hall, New York, 1996.