POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Correlated Topic Models

Course of
Bayesian Statistics

Group:
Francesca Anfossy
Anas Bahtaoui
Caspar Dietz
Kanthavel Pasupathipillai
Giulia Patanè
Théo Saulus

Advisor: Matteo Gianella
Co-advisors: Professor Alessandra Guglielmi
Academic Year: 2021-2022

Contents

1 | Overview of the project

1.1. Introduction and objectives

Topic modeling is part of a class of text analysis methods that analyze "bags" or groups of words together, instead of counting them individually, in order to capture how the meaning of words is dependent upon the broader context in which they are used in natural language.

One of the earliest topic models is Latent Dirichlet Allocation (LDA). It is one of the most popular topic modeling methods where each document is made up of various words, and each topic also has various words belonging to it. The aim of LDA is to find topics a document belongs to, based on the words in it.

One of the limitations of the LDA model is that it assumes independence between topics. However, that is not usually the case: for example a document about health and fitness is more likely to also be about food than a randomly chosen document.

To overcome such limitation, the Correlated Topic Model (CTM) was introduced. In this model the Dirichlet distribution used to model topic distribution in the LDA model is replaced with a logistic normal distribution which incorporates a covariance structure among the components and allows modeling of correlation between latent topics.

The representation of the relationships between latent topics can be modeled using graphs or graphical models in what we will refer to as topic graphs where each of the nodes of the graph represent a random variable (in this case a topic) and the existence of a edge denotes correlation between the two nodes.

The objective of this work is to study a way to learn the correlation structure between various topics of a set of documents using topic graphs, to learn the structure of the underlying topic graph of a group of documents. In order to do this, we aim at providing a Monte Carlo (MCMC) sampling strategy using both the CTM and graphical models.

2 | Theoretical overview

2.1. Correlated Topic Models

A topic model is a generative probabilistic model that uses a small number of distributions over a vocabulary to describe a document collection.

The correlated topic model (CTM) uses a more flexible distribution for the topic proportions that allows for covariance structure among the components. This gives a more realistic model of latent topic structure where the presence of one latent topic may be correlated with the presence of another.

2.1.1. CTM generative process

Notation

- Words and documents : Words are organized in documents. Let $w_{d,n}$ denote the n^{th} word in the d^{th} document. Each document is an element in a V-term vocabulary. And w_d denotes the vector of n_d words associated with document d.
- Topics : A topic β is a distribution over the vocabulary. The model will contain k topics $\beta_{1:k}$.
- Topic assignments : Each word is assumed drawn from one of the K topics. $z_{d,n}$ is the topic assignment associated with the n^{th} word and d^{th} document.
- Topic proportions : Each document is associated with a set of topic proportions θ_d which is a distribution over topic indices, and reflects the probabilities with which words are drawn from each topic in the collection.

The correlated topic model assumes that an n_d-words document d is generated via the following: Given k topics $\beta_{1:k}$, a k-vector μ and a $k \times k$ covariance matrix Σ that captures the dependencies among different topics:

1. Draw the *vector of topic prevalences* $\eta_d|\mu, \Sigma \sim \mathcal{N}(\mu, \Sigma)$
2. For n from 1 to n_d
   - Draw a *topic* assignment $Z_{d,n}|\eta_d$ from Mult(f(η_d)).
   - Draw a *word* $W_{d,n}|Z_{d,n}$ from Mult($\beta_{Z_{d,n}}$))

where θ is the *vector of topic proportions*

$$\theta = f(\eta) = \frac{\exp \eta}{\sum_i \exp(\eta_i)} \tag{2.1}$$

Hence we obtain a n_d-dimensional vector W_d, which is the collection of the words in the generated document d. [1]

2.2. Bayesian Gaussian graphical models

Graphical models provide powerful tools to uncover complicated patterns in multivariate data and are commonly used in Bayesian statistics and machine learning.

Graphical models are commonly used, particularly in Bayesian statistics and machine learning, to describe the conditional dependence relationships among variables in multivariate data.

In graphical models, each random variable is associated with a node and conditional dependence relationships among random variables are presented as a graph $G = (T, L)$ in which T specifies a set of nodes and L a set of existing links, i.e. a subset of $T \times T$.

We will work with undirected graphs. The absence of a link between two nodes specifies the pairwise conditional independence of those two variables given the remaining variables, while a link between them determines their conditional dependence. We work using undirected graphs meaning that $(i, j) \in L \iff (j, i) \in L$ for any pair $(i, j) \in T$. [3]

2.2.1. Model formulation

We can assume that:

$$\eta \sim \mathcal{N}_p\left(\mu, K^{-1}\right)$$

with K the precision matrix of the graph.

K is symmetric positive definite (for Σ to exist), and such that $(i, j) \in L \Leftrightarrow K_{\text{ij}} \neq 0$. Let us denote $P_G = \{K \ : K_{\text{ij}} \neq 0 \Leftrightarrow (i, j) \in L\}$

Hypothesis and likelihood

Let us assume that:

- $\mu = 0$
- $H = \left(\eta^{(1)}, \ldots, \eta^{(n)}\right)^T$ the topic prevalences for the observed data (computed, for example, from a corpus, *i.e.*, a set of documents)

Then, the likelihood is:

$$\mathcal{L}\left(H|K, G\right) \propto |K|^{\frac{n}{2}} \exp\left(-\frac{1}{2}\text{tr}\left(\text{KU}\right)\right) \tag{2.2}$$

with $U = H^T H$

Prior and posterior distribution of K

Let us choose a prior for K:

$$K \sim \text{G-Wishart}$$

Thus,

$$\mathbb{P}(K|G) = \frac{1}{I_G(b,D)}|K|^{\frac{b-2}{2}} \exp\left(-\frac{1}{2}\text{tr}\left(\text{DK}\right)\right)\mathbb{I}_{\{K\in P_G\}} \tag{2.3}$$

where:

- $b > 2$ is the degrees of freedom
- D is symmetric positive definite matrix
- $I_G(b,D)$ is a normalizing constant with respect to the graph G

It can be shown that this probability is the conjugate prior for the likelihood.

The posterior distribution is therefore:

$$\mathbb{P}(K|H,G) = \frac{1}{I_G\left(b^*,D^*\right)}|K|^{\frac{b^*-2}{2}} \exp\left(-\frac{1}{2}\text{tr}\left(D^*K\right)\right) \tag{2.4}$$

where:

- $b^* = b + n$
- $D^* = D + S$ that is G-Wishart(b^*, D^*)

2.2.2. Birth-Death MCMC sampling of K and G

We know that:

$$\mathbb{P}\left(K,G|H\right) \propto \mathbb{P}(K,G,H) = \mathbb{P}\left(H|K\right)\mathbb{P}\left(K|G\right)\mathbb{P}(G) \tag{2.5}$$

with

$$\mathbb{P}(G) \propto \left(\frac{\gamma}{1-\gamma}\right)^{|L|}$$

where:

- $|L|$ is the number of links
- $\gamma \in (0,1)$

Birth and Death of links

In order to explore all the possible links, we us a birth-death Markov process:

The death rate is $\delta_e(K)$ for $e \in L$, with:

$$\delta_e(K) = \min\left\{\frac{\mathbb{P}\left(G^{-e}, K^{-e}|H\right)}{\mathbb{P}\left(G, K|H\right)}, 1\right\}, \quad \text{and} \quad \delta(K) = \sum_{e\in L}\delta_e(K) \tag{2.6}$$

The birth rate is $\beta_e(K)$ for $e \notin L$, with:

$$\beta_e(K) = \min\left\{\frac{\mathbb{P}\left(G^{+e}, K^{+e}|H\right)}{\mathbb{P}\left(G, K|H\right)}, 1\right\}, \quad \text{and} \quad \beta(K) = \sum_{e\notin L}\beta_e(K) \tag{2.7}$$

The convergence of the MCMC to $\mathbb{P}\left(K, G|H\right)$ is therefore ensured.

Waiting time, to estimate the posterior graph probability:

$$W(K) = \frac{1}{\beta(K) + \delta(K)}$$

Birth-Death Algorithm

The BDMCMC algorithm samples a new adjacency matrix G by simulating births and deaths of links, and computing its associated waiting time. Afterwards, it samples a new precision matrix K associated to G sampling a G-Wishart. After some iterations, the most frequent graph G is chosen as the posterior expected correlation graph, as illustrated in the following figure.

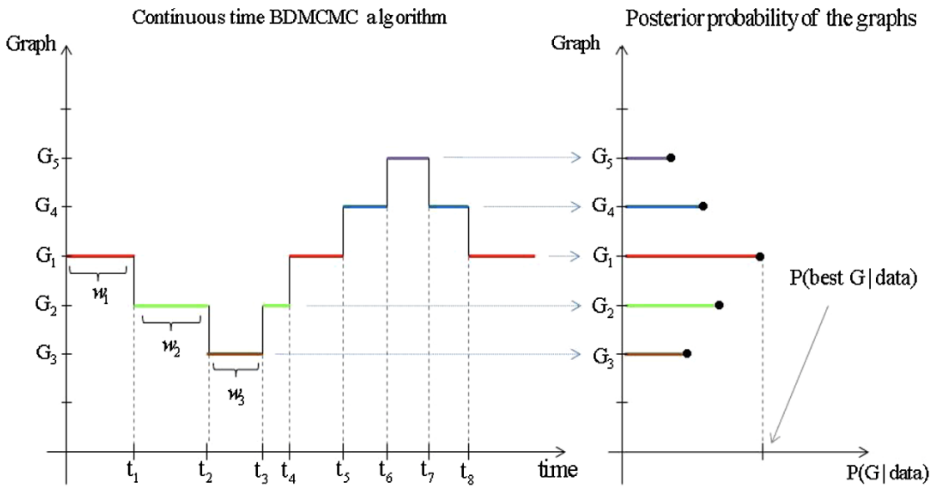

Figure 2.1: (Left) Continuous time BDMCMC algorithm with jumping and waiting times $\{(t_i, w_i)\}$. (Right) Posterior probability estimation of the graphs based on the proportions of their waiting times [2].

3 | Model Formulation

3.1. Notation

We consider the following notation

- Fixed parameters
  - D amount of documents
  - V size of the vocabulary
  - M maximum amount of repeated words in a document
  - k amount of topics
- Data
  - W matrix of $D \times V$ where $W_{d,n}$ is counter of appearances of the word n in document d
- Latent variables
  - Z matrix of $D \times V \times M$ where $Z_{d,n,m}$ is the topic index from which the m-th appearance of the word n on document d is drawn
  - B matrix of $k \times V$ where β_z is the parameter vector of the distribution for the z-th topic
  - C matrix of $k \times V$ where C_z is the V-dim vector of counts of sampled topics over each word for all documents
  - E matrix of $D \times k$ where E_d is the k-dim vector of counts of sampled drawings for the z-th topic over all words for each document
  - Θ matrix of $D \times k$ where θ_d is the k-dim vector that reflects the probability with which words are drawn from each topic in the collection
  - H matrix of $D \times k$ where η_d is the k-dim vector of the topic prevalences over document d
  - Σ matrix of $k \times k$ representing the covariance matrix
  - $G = (T, L)$ a k graph representing the links (edges $e \in L$) between topics (nodes $\in T$)
  - K matrix of $k \times k$ representing the precision matrix associated to the graph G

  - $\delta_e^{birth}, \delta_e^{death}$ the birth and death rates of edges e for the BDMCMC algorithm

- Transformations
  - Obtain C from Z:
$$C_z(i) = \sum_{d=1}^{D} \sum_{m=1}^{M} I(Z_{d,i,m} = z \tag{3.1}$$
  - Obtain E from Z:
$$E_d(i) = \sum_{n=1}^{V} \sum_{m=1}^{M} I(Z_{d,n,m} = i) \tag{3.2}$$
  - $\eta_d = \log(\theta_d)$ hence $\theta_d = exp\{\eta_d\}/(\sum_{d=1}^{D} exp\{\eta_d\})$
  - Obtain optimal birth rates from G, K for BDMCMC convergence $\delta_e^{birth}(K) = min\{P(G^{+e}, K^{+e}|Z)/P(G, K|Z), 1\} \forall e \notin L : G^{+e} = (V, L \cup \{e\}) \wedge K^{+e} \in \mathbb{P}_{G^{+e}}$
  - Obtain optimal death rates from G, K for BDMCMC convergence $\delta_e^{death}(K) = min\{P(G^{-e}, K^{-e}|Z)/P(G, K|Z), 1\} \forall e \in L : G^{-e} = (V, L \setminus \{e\}) \wedge K^{-e} \in \mathbb{P}_{G^{-e}}$

- Hyper parameters
  - μ vector of k that represents the mean of the prior distribution of each η
  - γ the Bernoulli probability of having a link between each pair of topics
  - b degrees of freedom of the prior G-Wishart distribution for K
  - S scale parameter, a symmetric positive definite matrix of the prior G-Wishart distribution for the precision matrix K

3.2. Model

We consider the adequate modelling as a combination of Correlated Topic Models and Graphical Models as follows: 3.3 and 3.4 follow the CTM model, hence topic and word wise levels, while 3.5 follows a graphical model. The transition point are the η vectors, that follow the CTM model but within a special case, satisfying conditions from the graphical model: a zero mean and the covariance matrix as the inverse of the precision matrix of the graphical part, a relation expected from theory.

$$W_{d,n} += 1 | Z_{d,n,m} \sim Mult(\beta_{Z_{d,n,m}}) \tag{3.3}$$

$$\beta \sim Dir(\alpha), \quad Z_{d,n,m}|\eta_d \sim Mult\left(\theta_d = \frac{exp\{\eta\}}{\Sigma_i \, exp\{\eta_i\}}\right) \tag{3.4}$$

$$\eta_d | \{\mu = 0, \Sigma = K^{-1}\} \sim N_k(\mu, \Sigma), \quad K \sim W_G(b, S) \quad , \quad G_{i \neq j} \sim Be(\gamma) \tag{3.5}$$

3.3. Gibbs Sampling algorithm

The algorithm is structured as follows

- Input
  - Initial guesses: Provide initial values for $B_0, Z_0, E_0, C_0, H_0, \Sigma_0, K_0, G_0$.
  - Data: W.
  - Fixed hyper parameters' values: Provide permanent values for μ, γ, b, S.
- Output
  - Sampling Z, B, H, Σ, G from their posterior distributions.
- Algorithm: For $i = 1 : J_{stop}$ iterate over the following steps
  1. MCMC sampling: Obtain $Z_{i+1}|W, \Theta_i, B_i$. Update E_{i+1}, C_{i+1}.
  2. Direct sampling: Obtain $B_{i+1}|W, C_{i+1}$.
  3. MH sampling: Obtain $H_{i+1}|K_i, E_{i+1}$. Update $\Theta_{i+1}|H_{i+1}$.
  4. BDMCMC sampling: Obtain $G_{i+1}, K_{i+1}|H_{i+1}$. Update $\Sigma_{i+1}|K_{i+1}$.

We will see in the next sections the detailed procedure of each sampling step.

4 | The algorithm in detail

4.1. Initialization

- Priors:
  - $B \sim Dirichlet(\alpha)$
  - $\eta_d | \{\mu, \Sigma\} \sim N_k(\mu, \Sigma)$ [1]
  - $Z_{d,n,m} | \eta_d \sim Mult\left(\theta_d = \frac{exp\{\eta\}}{\Sigma_i \, exp\{\eta_i\}}\right)$ [1]
  - $W_{d,n} += 1 | Z_{d,n,m} \sim Mult(\beta_{Z_{d,n,m}})$ [1]
  - $G \sim Bernoulli(\gamma)$ [3]
  - $K \sim W_G(b, S)$ G-Wishart distribution [3]
- Hyper parameters values:
  - k is fixed a priori
  - V is fixed after pre-processing the data
  - $M = \max_{i,j}\{W_{i,j}\}$ is fixed after pre-processing the data
  - $\mu = \vec{0}$ vector of zeros
  - $\alpha = \vec{1}$ vector of ones as non informative prior
  - $\gamma = 0.2$ a previously expected proportion of related topics with each other
  - $b = k - 1$ degrees of freedom for a $k \times k$ matrix
  - $S = I$ identity matrix as default scale
- Initial guesses:
  - As initial guess for B_0, we consider a matrix of ones, each vector normalized by the vocabulary size.
  - As initial guess for H_0, we consider a matrix of ones, each vector normalized by k, ant Θ is transformed accordingly.
  - As initial guess for K_0, Σ_0 we use the identity matrices.

- As initial guess for G_0 we transform it directly from K: $G_0(i \neq j) = 1 \iff K_0(i \neq j) \neq 0$ a zero-diagonal matrix indicating adjacency from K_0, hence it starts as a matrix of zeros.
- As initial guess for Z_0, we sample random topic indexes for the words in the observed W.

4.2. MCMC sampling of Z

This is inspired by the Collapsed Gibbs Sampling algorithm for the LDA approach [5].

We use the unnormalized posterior probability

$$p(Z_{dn} = z | W_{dn} = v, W_{\neg W_{dn}}, Z_{\neg Z_{dn}}, \Theta, B) \; \alpha \; (E_d(z) + \theta_d) \frac{C_z(v) + \beta_z}{\sum_{b \neq v} C_z(b) + V\beta_z}$$

The MCMC sampler algorithm is then

Algorithm 4.1 CGS algorithm

Input: W, Z, E, C
for $d \in \{1, \cdots, D\}$ **do**
  for $v \in \{1, \cdots, V\}$ **do**
    $\mathcal{I}_{di} \leftarrow W_{dv}$
    for $j \in \{1, \cdots, \mathcal{I}_{di}\}$ **do**
      $\hat{z} \leftarrow Z_{dvj}$
      $\rho \leftarrow array[k]$
      $E_d(\hat{z}) \leftarrow E_d(\hat{z}) - 1$
      $C_{\hat{z}}(v) \leftarrow C_{\hat{z}}(v) - 1$
      for $z = 1 : k$ **do**
        $\rho_z \leftarrow (E_d(z) + \theta_d) \times (C_z(v) + \beta_z)/(\sum_{b \neq v} C_z(b) + V\beta_z)$
      end for
      $\rho \leftarrow \frac{\rho}{||\rho||}$
      $\hat{z} \leftarrow Categorical(\rho))$
      $E_d(\hat{z}) \leftarrow E_d(\hat{z}) + 1$
      $C_{\hat{z}}(v) \leftarrow C_{\hat{z}}(v) + 1$
      $Z_{dvj} \leftarrow \hat{z}$
    end for
  end for
end for

Hence, this step is conformed by the following subalgorithm

1. Sample $Z_{i+1} \sim p\,(Z|W, \Theta, B)$
2. Update the E_{i+1}, C_{i+1} matrices by the transformations over Z_{i+1}

4.3. Posterior sampling of β_j's

We found that given as prior $\beta_j \sim Dirichlet(\alpha)$, its posterior distribution is $\beta_j \sim Dirichlet(\alpha + C_j)$, where C_j is j-th row of the matrix C.

$$\mathcal{L}(B|Z,W) \propto \prod_{d=1}^{D}\prod_{v=1}^{V}\prod_{r=1}^{M}\beta_{Z_{dvr}}(v)\prod_{j=1}^{k}\prod_{v=1}^{V}\beta_j^{\alpha_v-1}(v)$$

Then

$$\mathcal{L}(B|Z,W) = \mathcal{L}(B|C,W) = \mathcal{L}(B|C) \propto \prod_{j=1}^{k}\prod_{v=1}^{V}\beta_j^{\alpha_v-1+C_{jv}}(v)$$

Hence, in this step we obtain $B_{i+1}|C$ by direct sampling: For each topic z, we sample the vector $B_z \sim Dir(\alpha + C_z)$.

4.4. Metropolis-Hastings sampling

This step is obtaining for each document $\eta_d|E_d, \mu, \Sigma = K^{-1}$.

For a single document d the density of our target distribution is

$$g(\eta_d|E_d,\mu,\Sigma) \propto \frac{\exp\{-\frac{1}{2}\eta_d^T K\eta_d + \sum_{i=1}^{k} E_d(i)\eta_d(i)\}}{(\sum_{j=1}^{k} exp\{\eta_d(j)\})^k}$$

In order to sample a new η_d, both taking into account the document d and the prior distribution given by the covariance Σ, we can rely on Metropolis-Hastings method which allows this procedure provided a kernel for the posterior density.

We will use as proposal density of the MH algorithm a multivariate normal distribution centered in the current η_d with covariance equal to the identity matrix multiplied by a factor σ, with an adaptive approach [4]: at each iteration we multiply σ by an increasing factor (a number $a > 1$) when the proportion of accepted new rows of H is less than 0.44 (rule of thumb) otherwise by a decreasing factor ($a \in (0,1)$). To avoid instability we constrict σ in a limited range.

Then, we obtain $\Theta_{i+1} = \{\theta_{d,j}\}_{d,j}$, where $\theta_{d,j} = exp\{\eta_{d,j}\}/(\sum_{i=1}^{k} exp\{\eta_{d,i}\})$.

Hence, this step is conformed by the following sub-algorithm

1. Sample H_{i+1} from E_i and K_i
2. Obtain the Θ_{i+1} matrix by the transformation over H_{i+1}

4.5. BDMCMC sampling

The BDMCMC algorithm works in the following way:

Algorithm 4.2 BDMCMC algorithm

Input: $G = (T, L)$, K
for N iterations **do**
  Sample a graph
  Compute the $\delta_e(K)$, then compute $\delta(K)$
  Compute the $\beta_e(K)$, then compute $\beta(K)$
  Compute the $\beta_e(K)$, then compute $\beta(K)$
  Compute the waiting time $W(K) = \frac{1}{\beta(K)+\delta(K)}$
  Simulate a jump and update G
  Sample from the matrix K as $K \sim$ G-Wish(b, D)
end for

Hence we use this step in the following way

1. Update first G
2. Update then K
3. From the couple (G, K), we keep the precision matrix for $\Sigma = K^{-1}$

5 | Conclusions

We implement the generating and sampling process as stated in the previous chapter, in order to test with artificial data of different sizes and assess the results. As illustrated and comparing the generating process with the sampling process (see the following figure), we can highlight that the implementation works within the same logic. The generating process samples K, B, H and Z, while the sampling goes backwards.

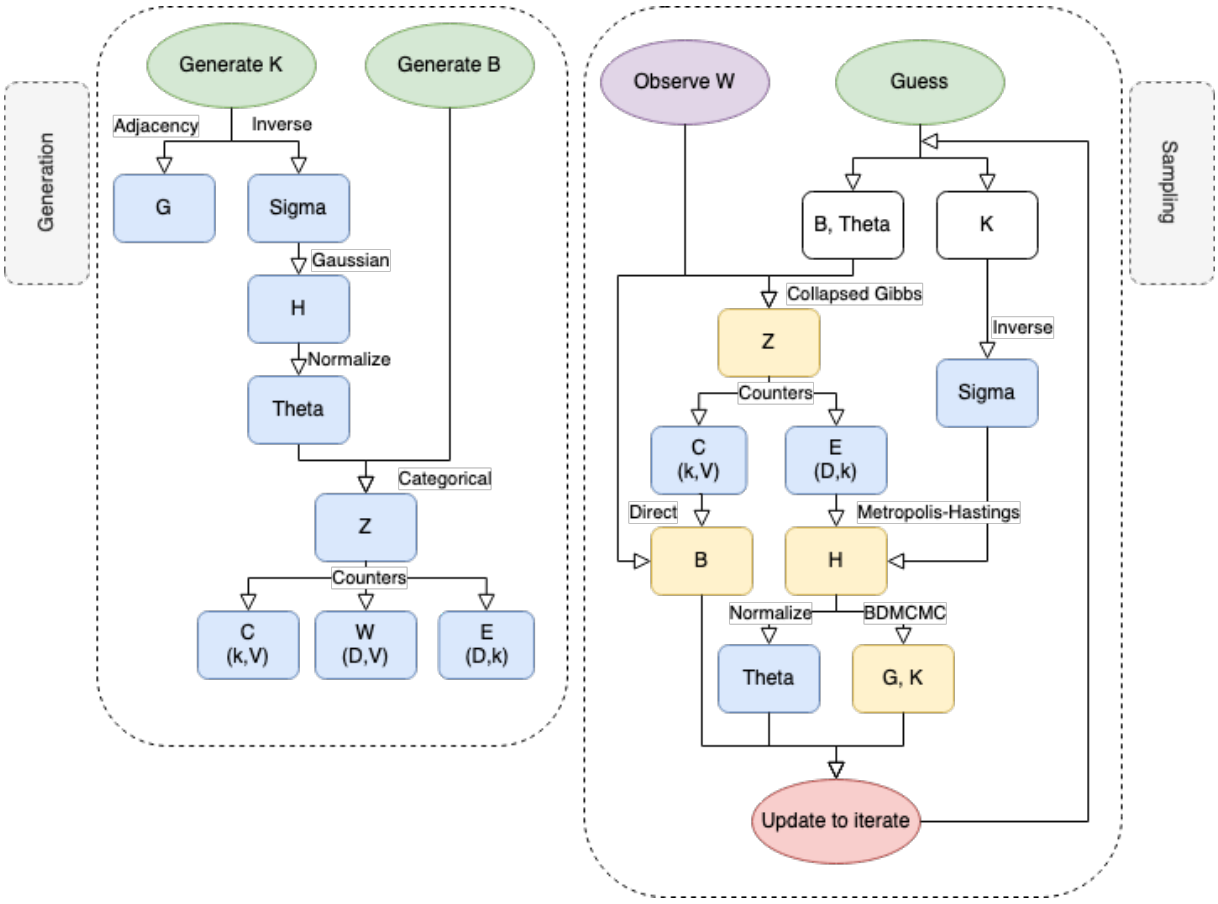

Figure 5.1: Comparison between the generating process (left) and the sampling process (right).

We implement both processes in Python, with the BDMCMC sampling using the R package `BDgraph` in a subprocess, and assess the results by stages and verifying the output.

5.1. Individual samplers

In order to test convergence and stability of each part of the global sampler, for each sampling step we do the following

1. We assume as fixed parameters the values of the target latent variables that the specific sampler uses as input.
2. We do a cycle of 4000 iterations where only the output values are updated.
3. We compute error metrics, according to the type of object, with respect to the target value, in order to look at the error series and other descriptors of it.

For the Z sampler, we look at the binder loss function of Z against its target value, since it's a clustering object that contains topic indexes as values. Since the binder loss is a positive sum of integers, we verify convergence and stability by seeing a histogram skewed to the left, while still having a series with a caterpillar shape. We confirm both behaviors, hence the sampler is verified.

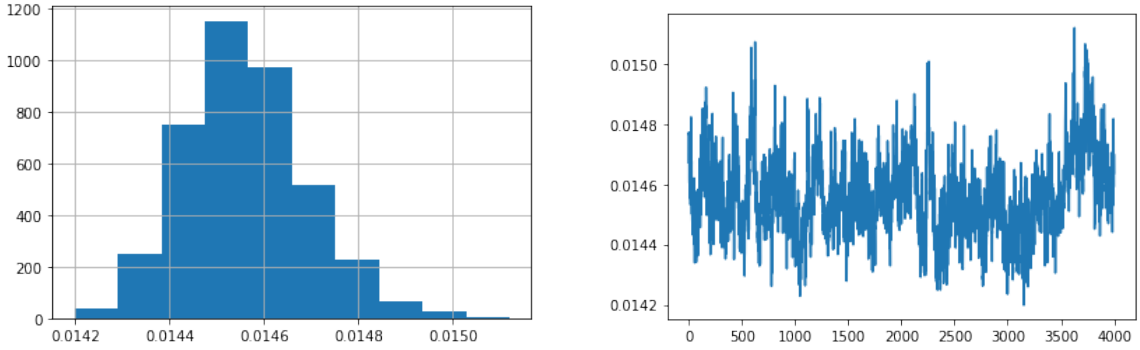

Figure 5.2: Histogram and series of Binder loss for Z.

For the B sampler, since its vectors have continuous values, we look at the L2 error against its target value, for each topic. This means we compare each β as vector, hence we obtain a different series for each topic. Since the L2 error is also always positive, we verify convergence and stability by seeing a histogram skewed to the left, while still having a series with a caterpillar shape. We confirm both behaviors, hence the sampler is verified.

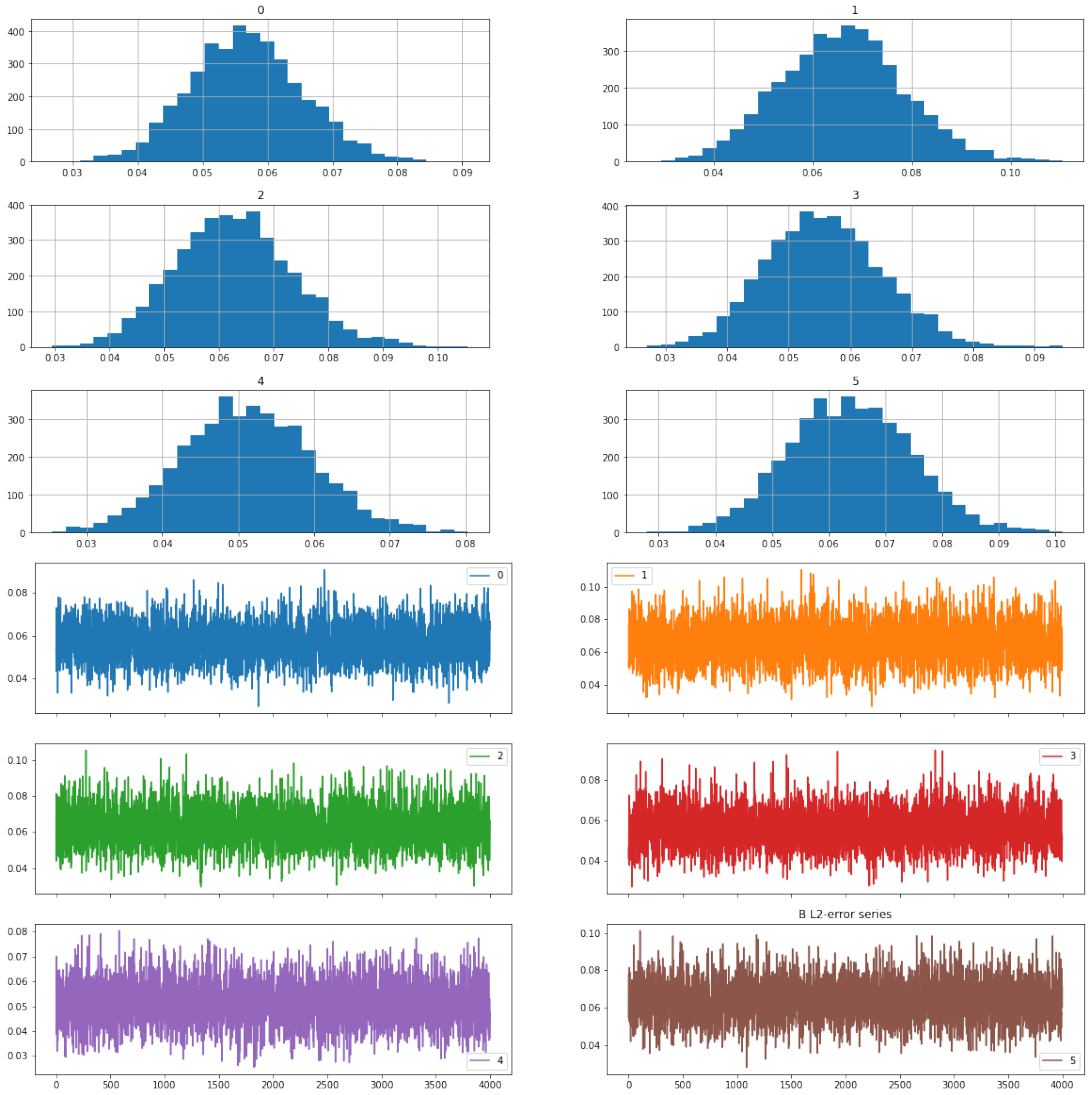

Figure 5.3: Histogram and series of L2 loss for B, by topic.

For the H sampler, we assess visually the output through heat maps, where we compare the sampled Θ (its deterministic transformation) and the target. The objective is to be able to see the high concentrations of some topics in the resulting matrix by comparing the vectors. Since we distinguish they are captured by the sampler, we consider it also verified.

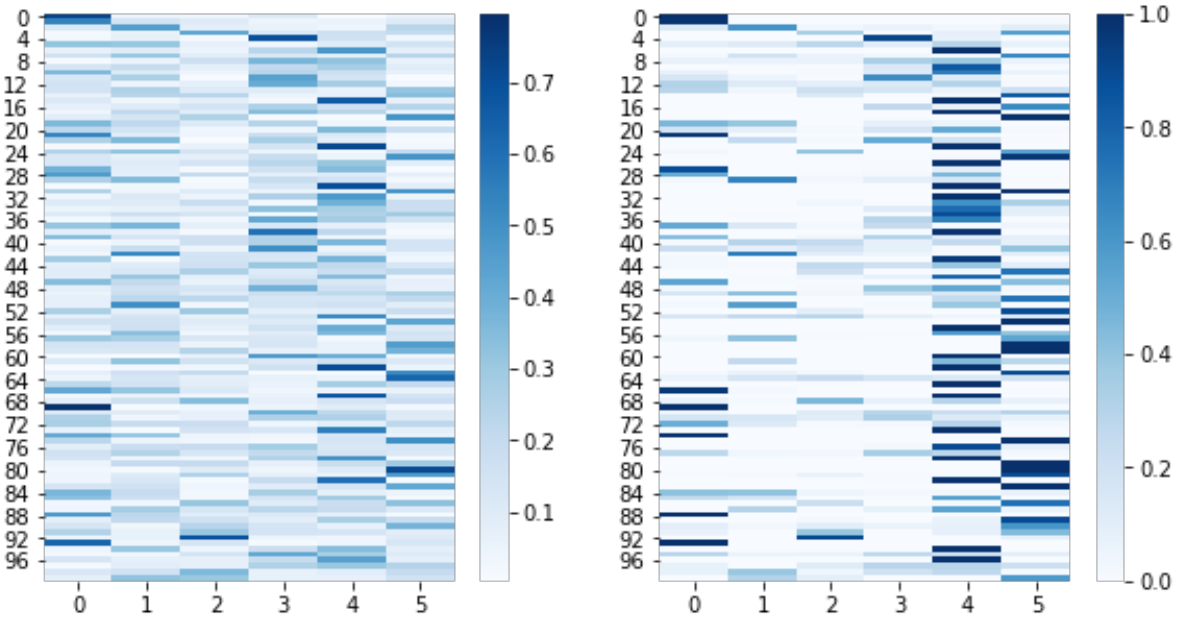

Figure 5.4: Heatmap comparison between the true Θ (left) and the last sampled one (right).

For the graphical part of the sampler, we consider different metrics for each object. For G, the adjacency matrix, we count the mismatches among edges. For K, the precision matrix, we consider its transformation Σ, the covariance matrix, to which we evaluate with the L2 error against its target.

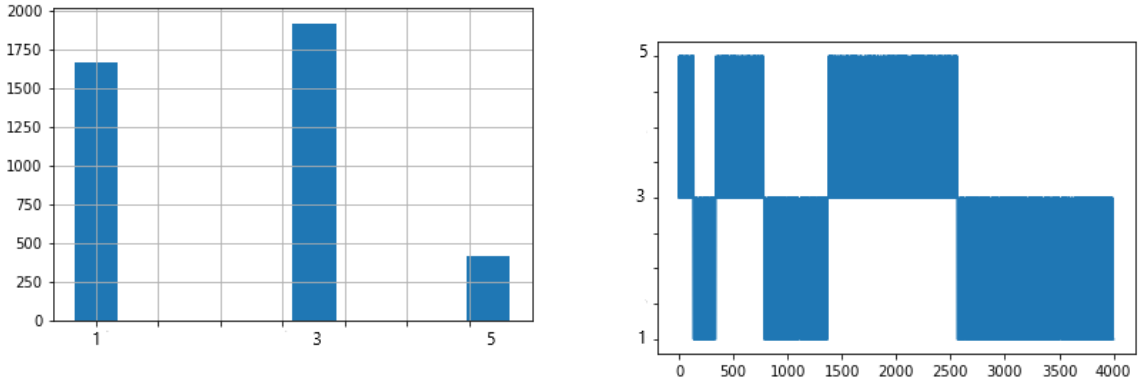

Figure 5.5: Histogram and series of wrong edges allocation (15 possible edges) for G.

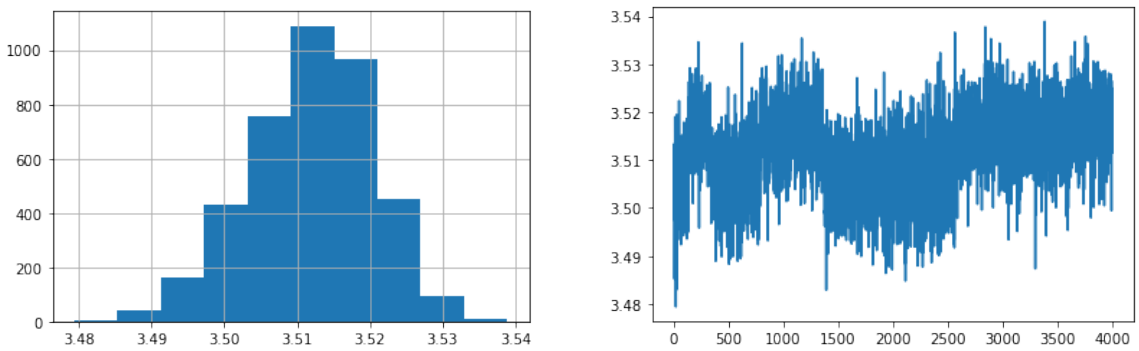

Figure 5.6: Histogram and series of L2 loss evolution for K.

5.2. Global sampling

We now present the results of the application case for the complete cycle, with target data made artificially and randomly. As initial guesses, we use the uninformative set of latent variables mentioned in the previous chapter, in order to have a symmetric starting point, with exception of Z, E, and C, that are randomly sampled from the observed W.

In 5000 iterations, we perform the following

- Compute the binder loss for each Z with respect to its target, and with respect to its previous iteration.
- Compute the L2 difference for each iteration between the new sampled B and (1) its target, and (2) the previous iteration.
- Compute the graph loss for each G matrix and the L2 error for each Σ matrix, with respect to their target.
- Select the learned graph, discarding the samples before the burn in of 1000 iterations.

For analysis, we report different criteria for selecting the graph. Some coincide, while others can lead to far results from the target. Hence, the criteria can be a key point to consider the learned output. Furthermore, we verify that the latent variables of interest have a consistent behavior, since each converge to specific neighborhoods of their respective space. We display part of the results for the steps performed as mentioned.

For what concerns the topic distributions (B) and their correlation scheme (G), we need to highlight the following considerations on B:

- In the global sampler the topics arises in **different order**
- We cannot directly compare the sampled B matrix with the true one: we need to identify the permutation of rows for which the topics in the same row are the most similar. We used the algorithm of stable matching to associate them.
- Once computed the **permutation**, we compare the sampled G with its target.

However, the permutations don't necessarily stabilize, so there is not a clear correspondence among the true topics and the sampled ones. The **graphic loss** takes as input the target graph, the sampled one and the permutation computed on their corresponding B; it returns the number of wrong edges.

Comparing graphs by frequency leads to a learned one that has sharply higher visits than the other candidates, so we can say that converges to that graph. By contrast, we see that the accumulated waiting times can lead to ties and the chosen graph can be sensitive to the burn in.

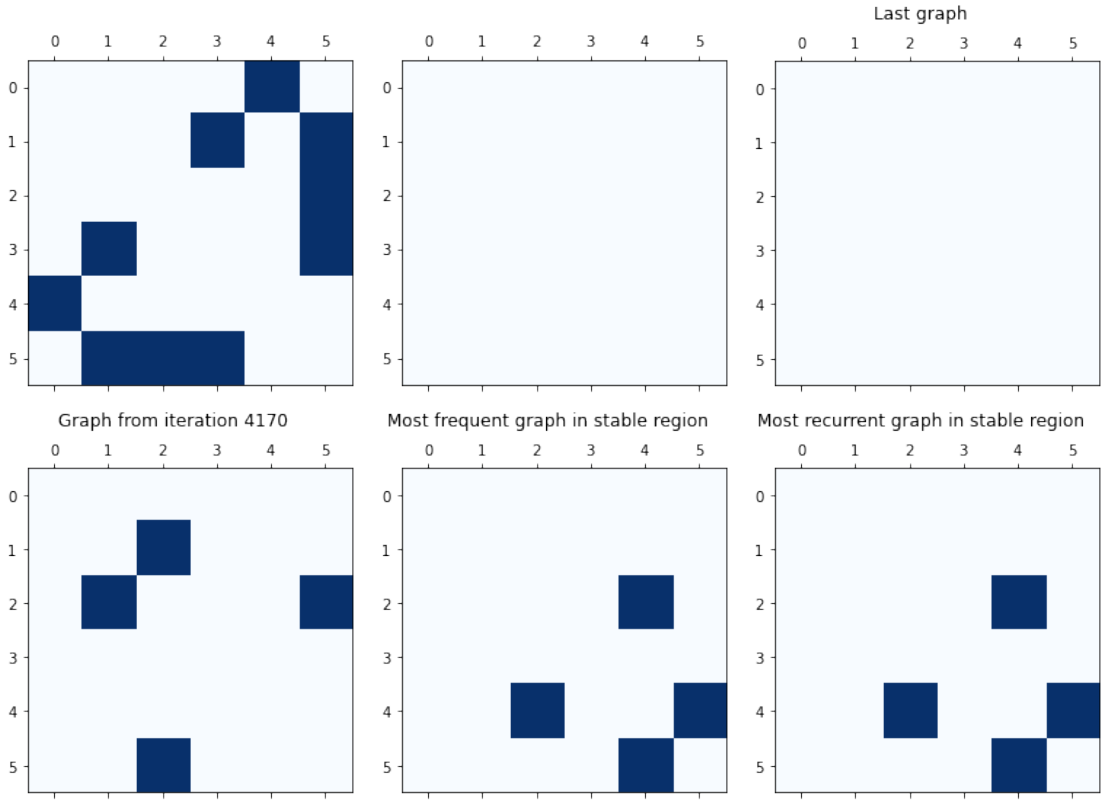

Figure 5.7: G matrix from top-left to bottom-right: Target, initial guess, last sampled, lowest binder loss on Z, highest waiting time, most sampled.

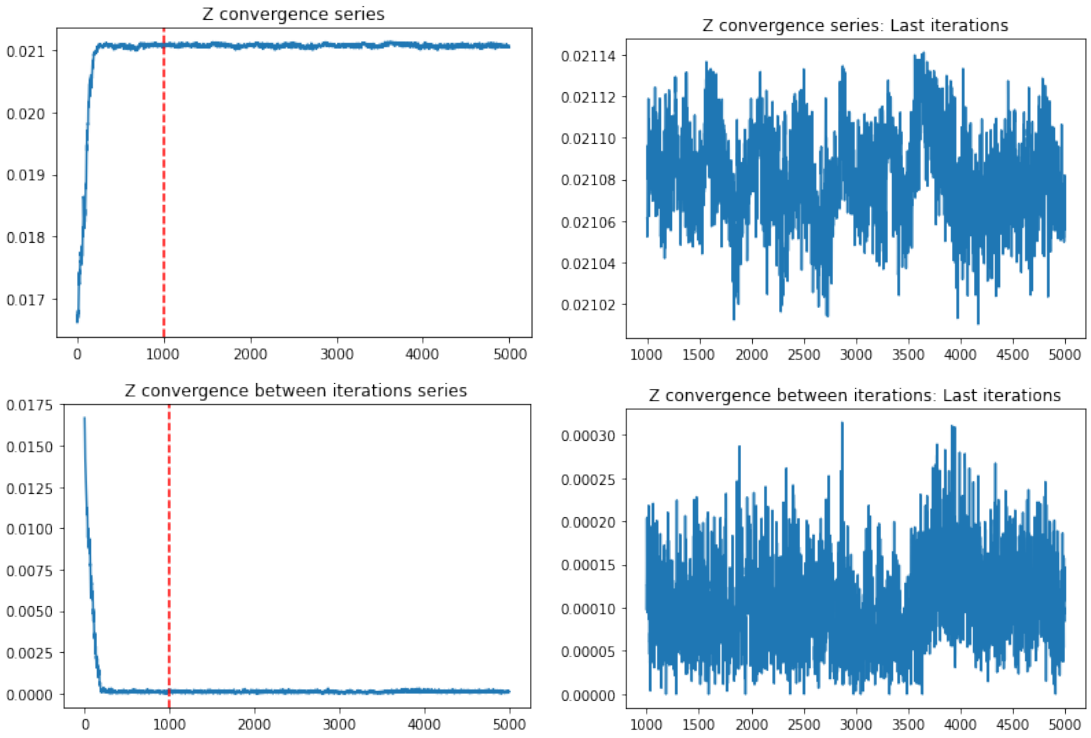

Figure 5.8: Binder loss for Z with respect to (top) its target and (bottom) the previous iteration, (left) with and (right) without burn in.

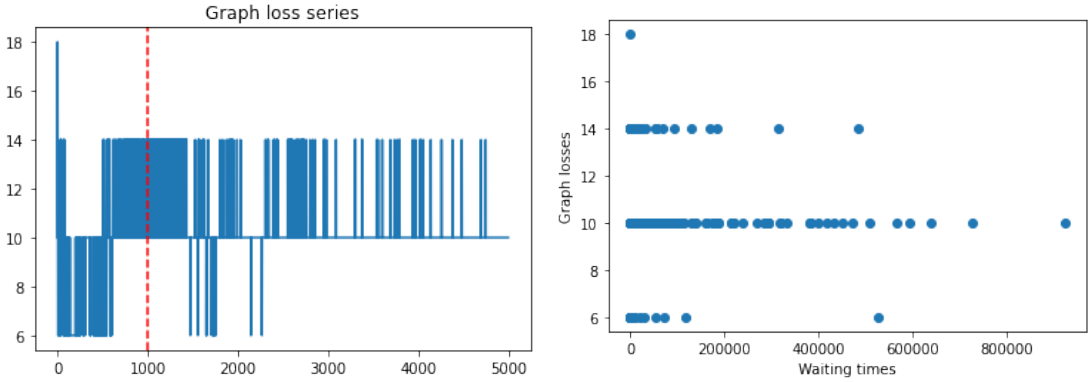

Figure 5.9: Graph losses series and scatter plot against waiting times.

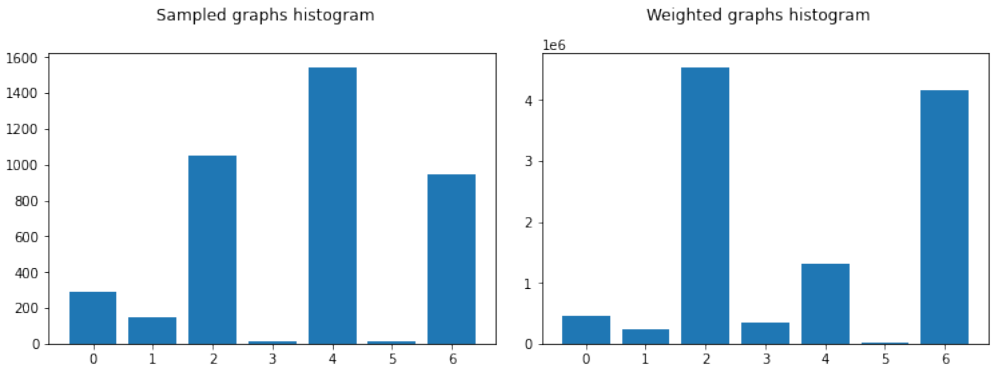

Figure 5.10: Graphs weights after the burn in: By amount of sample repetitions and by accumulated waiting times.

5.3. Final remarks

In our implementation, we find that the individual samplers reach the target distributions. However, in the global sampler, it is difficult to assess the convergence and we think we should still explore:

- Other random initial guesses
- Other type of adaptive methods
- Different values of burn in
- Reformulate the error measures

Bibliography

[1] David Blei and John Lafferty. "Correlated Topic Models". In: *NIPS '06: Advances in Neural Information Processing Systems* 18 (Jan. 2005).

[2] Abdolreza Mohammadi and Ernst C Wit. "Bayesian structure learning in sparse Gaussian graphical models". In: *Bayesian Analysis* 10.1 (2015), pp. 109–138.

[3] Reza Mohammadi and Ernst C. Wit. "BDgraph: An RPackage for Bayesian Structure Learning in Graphical Models". In: *Journal of Statistical Software* (2019). URL: `https://arxiv.org/pdf/1501.05108.pdf`.

[4] Gareth O. Roberts and Jeffrey S. Rosenthal. "Examples of Adaptive MCMC". In: *Taylor Francis* (June 2009), pp. 353, 356.

[5] Han Xiao and Thomas Stibor. "Efficient Collapsed Gibbs Sampling For Latent Dirichlet Allocation". In: *JMLR: Workshop and Conference Proceedings* 13 (Nov. 2010), pp. 63–78.