Calofir Ionut

Homework 1
-LDA-

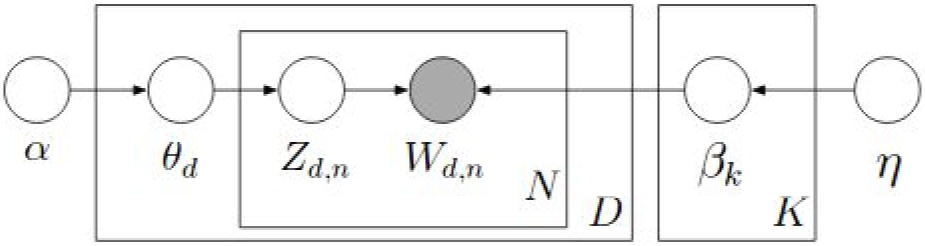
Latent Dirichlet Allocation is a generative statistical model that allows to assign a mixture of topics to a set of documents.

1. Data Processing

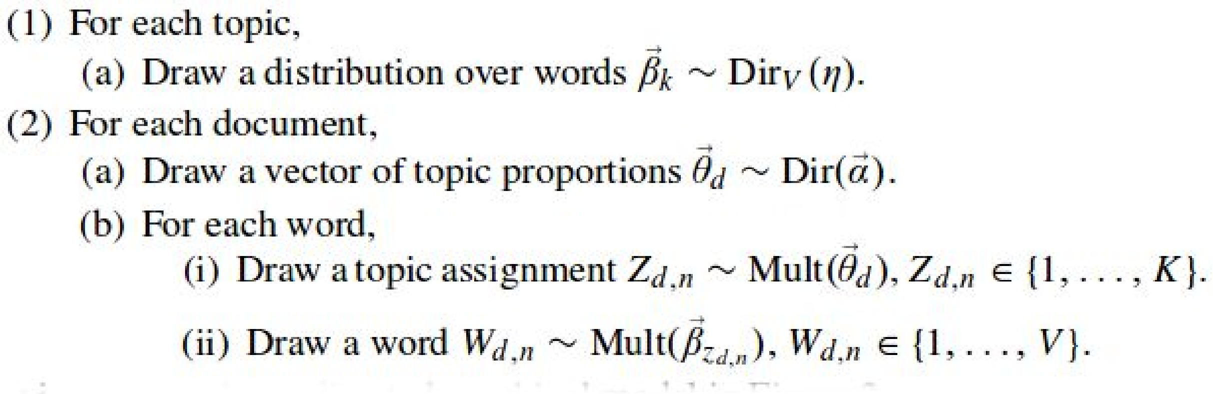
For this task I preprocessed the documents in the following way:

- I put each document on **a line** in a file text
- I read the file line by line, and for every document:
    - I **tokenized** the document by words
    - I applied **stanford POS Tagger**
    - I kept only the **nouns** and **verbs**
    - I applied **lemmatization**
    - I removed **stopwords**
    - I removed words smaller than **2 characters**
    - I remove words that contains **numbers**
    - I converted words to **lowercase**

2. LDA Basic



LDA model (image from **Topic Models** by **David M. Blei** and **John D. Lafferty**)



```
(1) For each topic,
    (a) Draw a distribution over words $\vec{\beta}_k \sim \mathrm{Dir}_V(\eta)$.
(2) For each document,
    (a) Draw a vector of topic proportions $\vec{\theta}_d \sim \mathrm{Dir}(\vec{\alpha})$.
    (b) For each word,
        (i) Draw a topic assignment $Z_{d,n} \sim \mathrm{Mult}(\vec{\theta}_d)$, $Z_{d,n} \in \{1, \ldots, K\}$.
        (ii) Draw a word $W_{d,n} \sim \mathrm{Mult}(\vec{\beta}_{z_{d,n}})$, $W_{d,n} \in \{1, \ldots, V\}$.
```

LDA algorithm (image from **Topic Models** by **David M. Blei** and **John D. Lafferty**)

Notations: In my code, **ETA** is **BETA** and **BETA** is **PHI**.

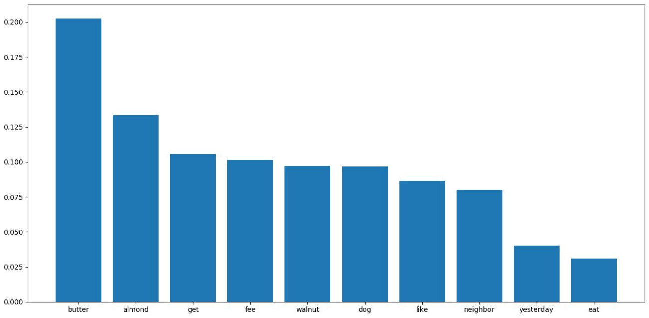
For this task I implemented the model with the above distributions. My implementation can be found in **lda_model.py(_mcmc_lda)**. I used the following parameters: topics: 2, number of iterations: 40000, burn-in: 10000, thin: 1.

Here are some results and also can be found in results/basic:

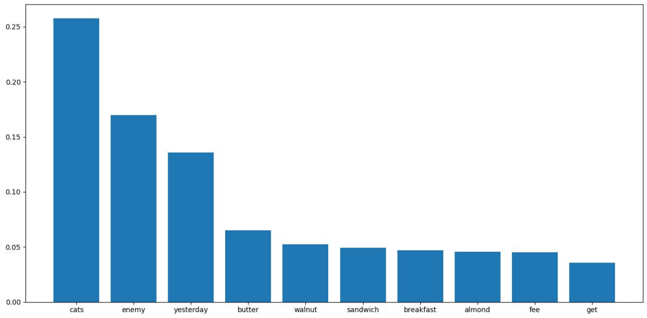
1. Dummy text

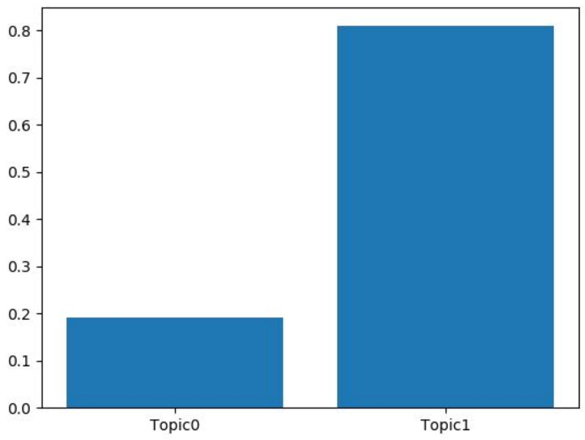
   Text: [I had a peanuts butter sandwich for breakfast.
   I like to eat almonds, peanuts and walnuts.
   My neighbor got a little dog yesterday.
   Cats and dogs are mortal enemies.
   You mustn't feed peanuts to your dog.]

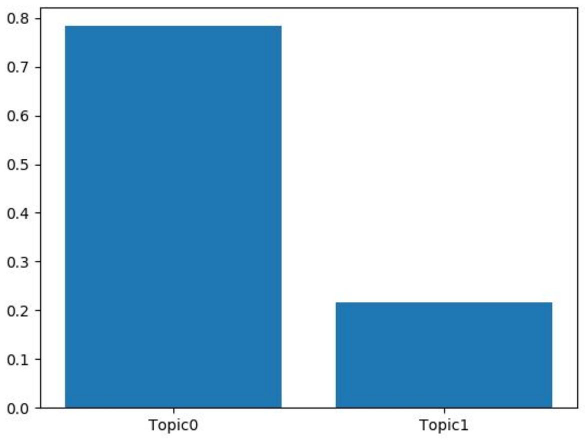
Topic 1

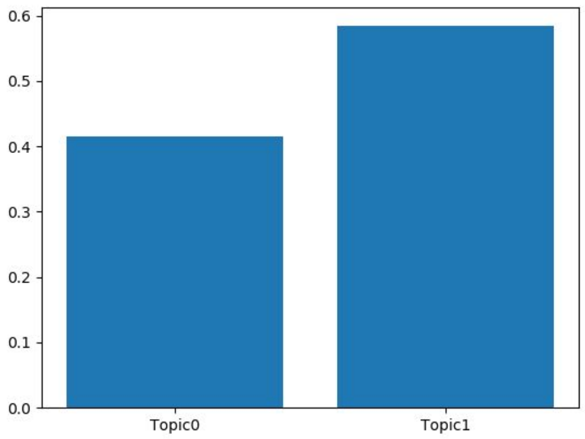


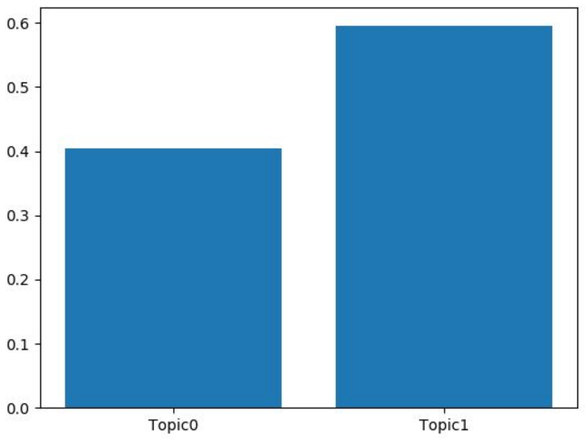
Topic 2

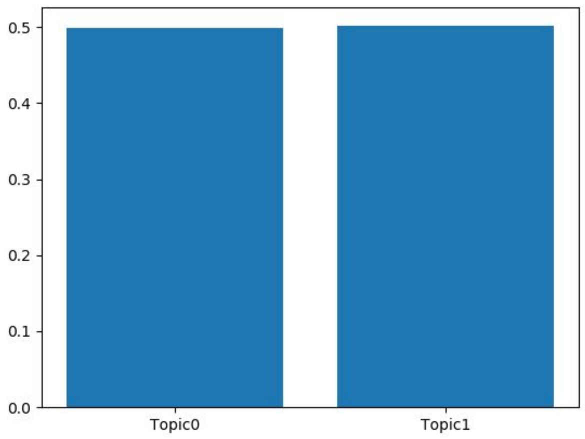



0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Topic0
Topic1


0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Topic0
Topic1

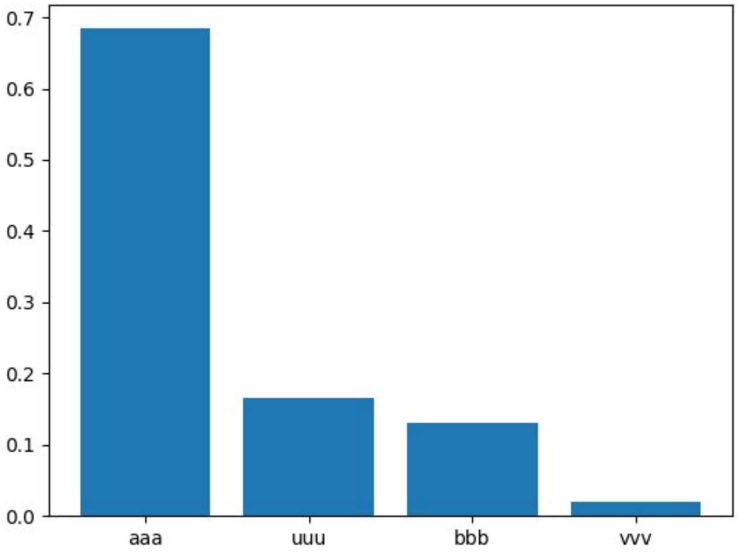

0.0
0.1
0.2
0.3
0.4
0.5
0.6
Topic0
Topic1

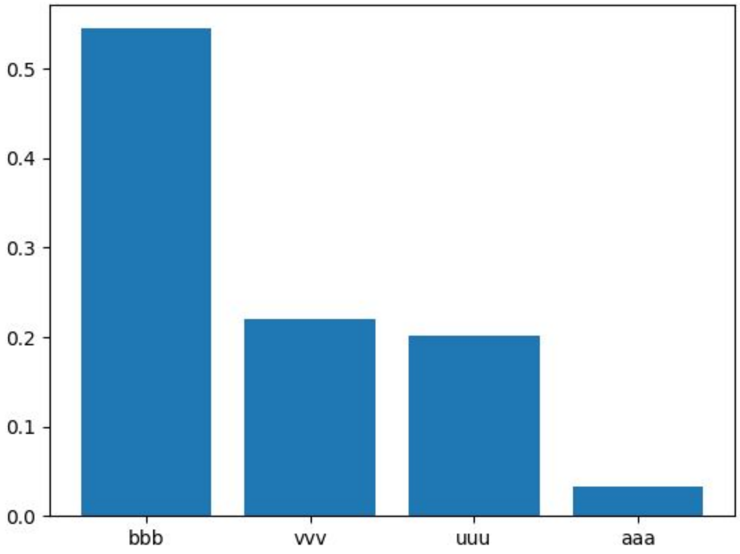

0.0
0.1
0.2
0.3
0.4
0.5
0.6
Topic0
Topic1


0.0
0.1
0.2
0.3
0.4
0.5
Topic0
Topic1

2. Dummy text2

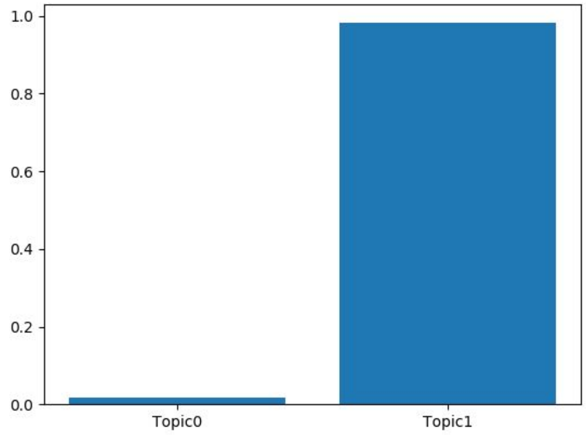
   Text: [aaa bbb aaa
   bbb aaa bbb
   aaa bbb bbb aaa
   uuu vvv
   uuu vvv vvv
   uuu vvv vvv uuu]

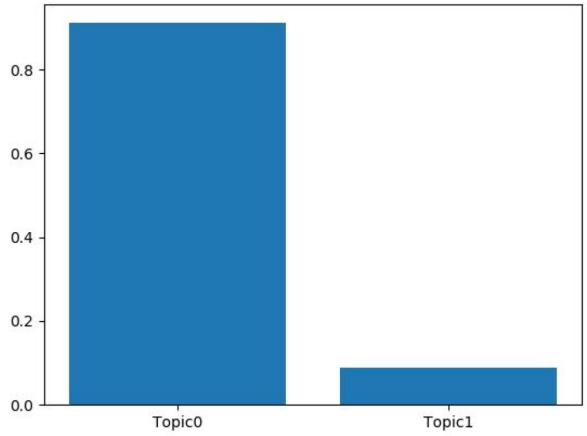


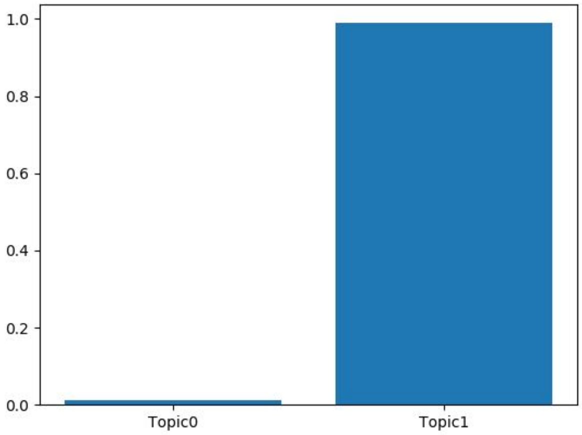


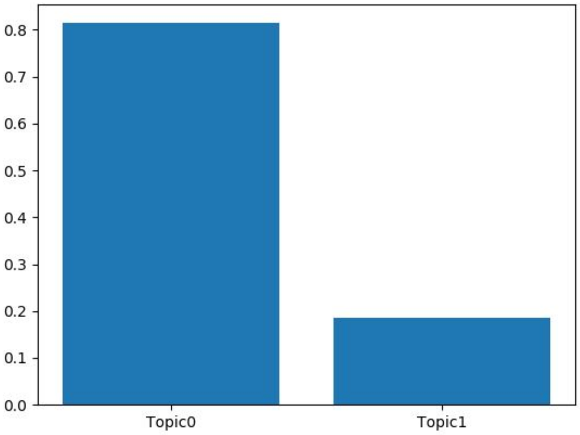
Topic 1 (above) and Document 1(below)

Topic 2 (above) and Document 2 (below)



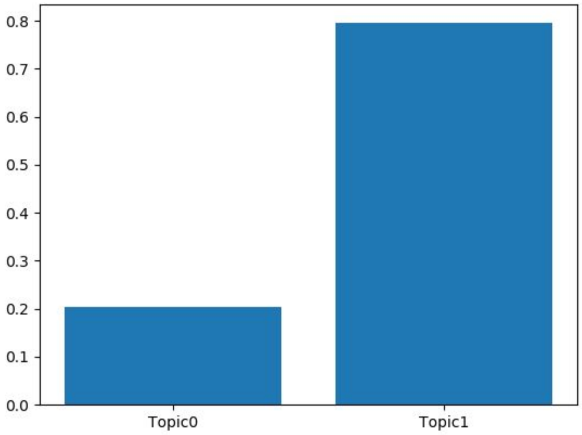


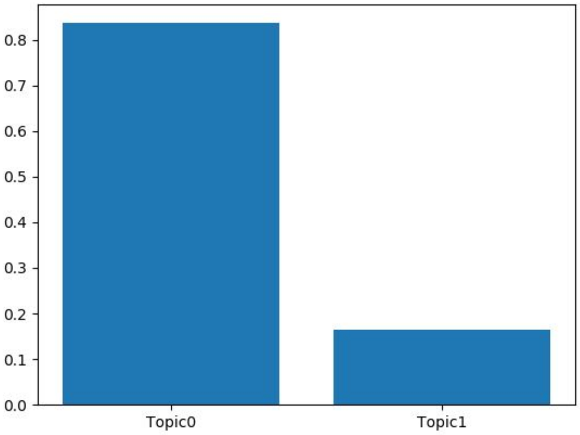




Document 3 (above) and 5 (below)

Document 4 (above) and 6 (below





3. Real text

I tried to apply the LDA model on real text. So I downloaded two articles about neuroscience and two abstracts from arxiv about machine learning. This text can be found in data/basic/train_data_real.txt (the first 2 texts are about neuroscience and the other 2 are about machine learning). Also, to plots can be found in plots/basic/text_real.

3. Similarity

For topic-based similarity measure between two documents I used Hellinger distance described in the paper **Topic Models** by **David M. Blei** and **John D. Lafferty**.

$$\text{document-similarity}_{d,f} = \sum_{k=1}^{K} \left(\sqrt{\widehat{\theta}_{d,k}} - \sqrt{\widehat{\theta}_{f,k}} \right)^2$$

Hellinger Distance (image from **Topic Models** by **David M. Blei** and **John D. Lafferty**)

Here are some results for the **Dummy text** from **2. LDA basic**:

```
[[0.00000000e+00 3.89497735e-01 6.16795737e-02 5.63138235e-02
  1.09281112e-01]
 [3.89497735e-01 0.00000000e+00 1.46930106e-01 1.55419695e-01
  9.08680646e-02]
 [6.16795737e-02 1.46930106e-01 0.00000000e+00 1.23892330e-04
  6.90238924e-03]
 [5.63138235e-02 1.55419695e-01 1.23892330e-04 0.00000000e+00
  8.87371870e-03]
 [1.09281112e-01 9.08680646e-02 6.90238924e-03 8.87371870e-03
  0.00000000e+00]]
```

4. How to handle new documents?

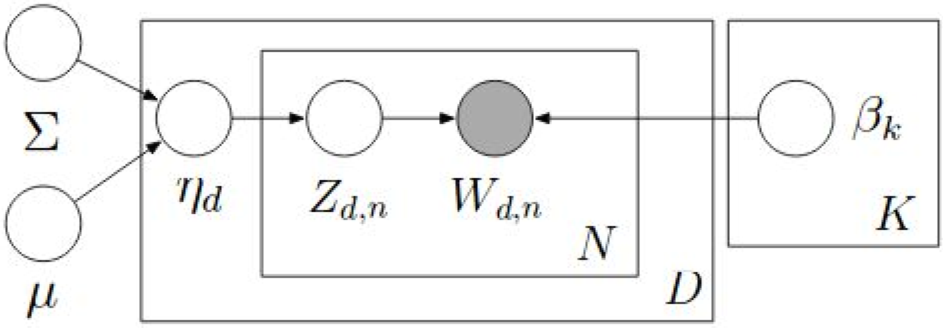
One solution is to append the new document to the old corpus and retrain the model. But this approach will change the distributions of the old corpus so in my opinion is not that good.

In this homework my approach was to compute the **P(topic | word)** and from this distribution to draw a random topic for each word. **P(word |**

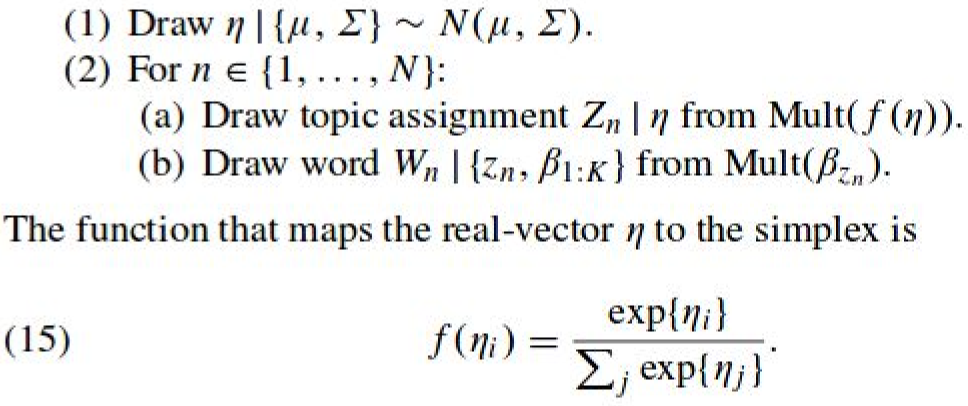
topic) is given by **PHI**, **P(word)** can be computed from the trained corpus and **P(topic)** can be computed from **Z**.

Text: [I had a peanuts butter sandwich for breakfast.
I like dogs because they also like peanuts butter sandwich.]
Results: [[0, 0, 1, 0]
[0, 1, 1, 0, 1, 1]]

5. LDA Correlated



LDA correlated model (image from **Topic Models** by **David M. Blei** and **John D. Lafferty**)



(1) Draw $\eta \mid \{\mu, \Sigma\} \sim N(\mu, \Sigma)$.
(2) For $n \in \{1, \ldots, N\}$:
(a) Draw topic assignment $Z_n \mid \eta$ from Mult($f(\eta)$).
(b) Draw word $W_n \mid \{z_n, \beta_{1:K}\}$ from Mult(β_{z_n}).

The function that maps the real-vector η to the simplex is

(15) $$f(\eta_i) = \frac{\exp\{\eta_i\}}{\sum_j \exp\{\eta_j\}}.$$

LDA correlated algorithm (image from **Topic Models** by **David M. Blei** and **John D. Lafferty**)

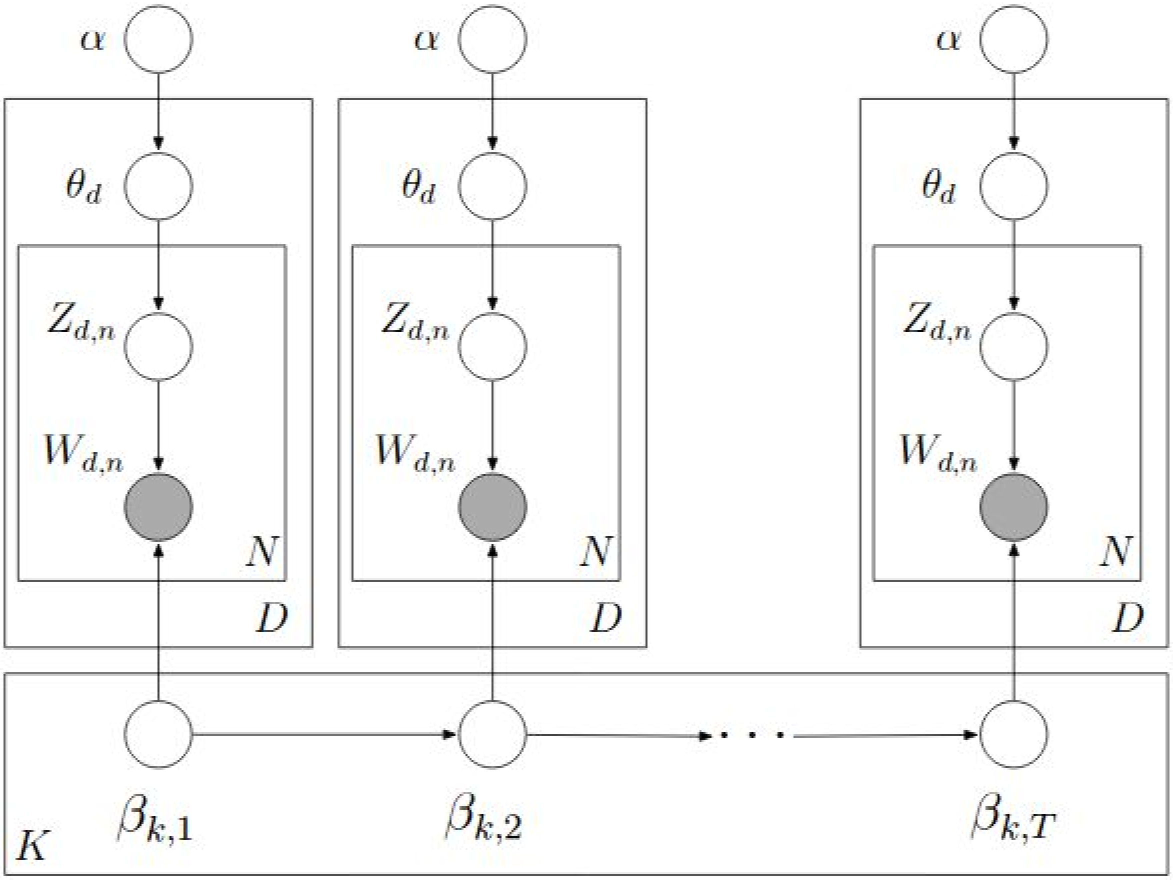
Notations: In my code, **ETA** is **BETA**, **BETA** is **PHI** and **SIGMA** is **TAU**.

To compute the precision matrix I built a variable with a **Wishart** distribution. I used the following parameters: topics: 2, number of iterations: 20000, burn-in: 5000, thin: 1.

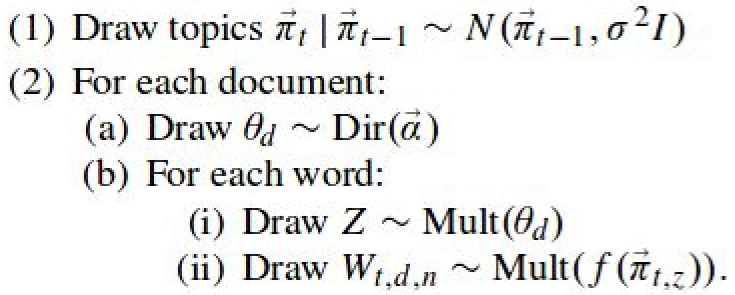
The correlated LDA model is based on the idea that in documents, some topics tends to appear together.

Results: In my homework, I applied this model on the text from **2.3** and the plots can be found in plots/correlated/text_real.

6. LDA Dynamic



LDA dynamic model (image from **Topic Models** by **David M. Blei** and **John D. Lafferty**)



```
(1) Draw topics $\vec{\pi}_t \mid \vec{\pi}_{t-1} \sim N(\vec{\pi}_{t-1}, \sigma^2 I)$
(2) For each document:
    (a) Draw $\theta_d \sim \mathrm{Dir}(\vec{\alpha})$
    (b) For each word:
        (i) Draw $Z \sim \mathrm{Mult}(\theta_d)$
        (ii) Draw $W_{t,d,n} \sim \mathrm{Mult}(f(\vec{\pi}_{t,z}))$.
```

LDA dynamic algorithm (image from **Topic Models** by **David M. Blei** and **John D. Lafferty**)

Notations: In my code, **ETA** is **BETA** and **BETA** is **PHI**.

To compute pi[0] I built a variable with a **Uniform** distribution. I used the following parameters: topics: 2, number of iterations: 10000, burn-in: 1000, thin: 1.

The dynamic model is based on the idea that over time, some fields can change and so the topics evolve.

Results: In my homework, I downloaded 2 abstracts about machine learning from arxiv from years 2013-2014 to test the dynamic LDA model. Plots per year to see how the topics changed can be found in plots/dynamic/tex_real.

7. Conclusions

In this homework I built the LDA model and tried different approaches to improve the model.