

## Inference and Representation, Fall 2015

### Problem Set 4: Gibbs sampling

**Due: Friday, October 23, 2015 at 5pm** (uploaded to NYU Classes.)

**Your submission should include a PDF file called “solutions.pdf” with your written solutions, separate output files, and all of the code that you wrote.**

**Important:** See problem set policy on the course web site.

Latent Dirichlet allocation (LDA) is a probabilistic model for discovering topics in sets of documents [1]. The generative model is as follows:

- For each document,  $m = 1, \dots, M$ 
  1. Draw topic probabilities  $\theta_m \sim p(\theta|\alpha)$
  2. For each of the  $N$  words:
    - (a) Draw a topic  $z_{mn} \sim p(z|\theta_m)$
    - (b) Draw a word  $w_{mn} \sim p(w|z_{mn}, \beta)$ ,

where  $p(\theta|\alpha)$  is a Dirichlet distribution, and where  $p(z|\theta_m)$  and  $p(w|z_{mn}, \beta)$  are Multinomial distributions. Treat  $\alpha$  and  $\beta$  as fixed hyperparameters. Note that  $\beta$  is a matrix, with one column per topic, and the Multinomial variable  $z_{mn}$  selects one of the columns of  $\beta$  to yield multinomial probabilities for  $w_{mn}$ .

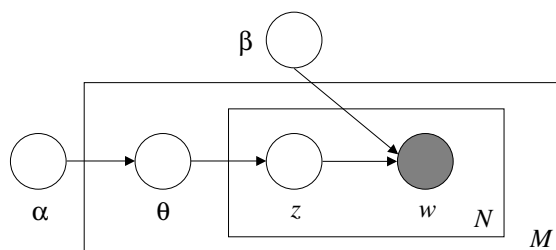


Figure 1: Graphical structure of the LDA model.

1. In this question you will use an off-the-shelf implementation of LDA to get practice with learning topic models on real-world data, and to analyze various trade-offs that can be made during learning.
  - (a) Prepare a corpus of documents from which you'll learn. You can find some already prepared text collections here:  
<https://archive.ics.uci.edu/ml/datasets/Bag+of+Words>  
 However, we prefer that you be creative and construct your own!
  - (b) Learn a latent Dirichlet allocation model on your corpus using default parameters. You can use any software package that you like. Two excellent options are:

- Mallet (<http://mallet.cs.umass.edu/>)
- Gensim (<http://radimrehurek.com/gensim/>)

Qualitatively describe what topics are discovered.

- (c) Re-run learning using varying numbers of topics (e.g., 5, 20, 100). Describe qualitatively the differences that you observe as the number of topics increases.
2. Derive a Gibbs sampler for the LDA model (i.e., write down the set of conditional probabilities for the sampler; see Sec. 24.2 of Murphy). To obtain full credit, you must hand in your full derivation, not just the final formulas.
- You may find it helpful to refer to your solutions from Problem Set 2 (question 4).*
3. Derive a collapsed Gibbs sampler for the LDA model, where you consider the marginal distribution  $\Pr(\mathbf{z}_m \mid \mathbf{w}_m; \alpha, \beta)$  (integrating out the topic probabilities  $\theta_m$ ) and are now only sampling  $\mathbf{z}$ . Again, you must hand in your full derivation.
4. Implement both of the inference algorithms that you derived. You will then run your algorithms to find the posterior topic distribution  $\theta$  for an input document.

We have previously learned the parameters (i.e.,  $\alpha$  and  $\beta$ ) of a 200-topic LDA model on a corpus containing thousands of abstracts of papers from the top machine learning conference, Neural Information Processing Systems (NIPS). Your task will be to infer the topic distribution for a new document.

We have provided the following data files:

- `alphas.txt`, which has on each line for topic  $i$ :  $i$ ,  $\alpha_i$ , and a list of the most likely words for this topic,
- `abstract_*.txt`, with the words of document  $m$  (i.e., the abstract),
- `abstract_*.txt.ready`, with, in order,
  - the number of topics  $k$ ,
  - $\alpha_i$ , for  $i = 1, \dots, k$ ,
  - for every word  $w_n$ , the word itself followed by  $\beta_{w_n, i}$  for  $i = 1, \dots, k$ .

Note that your code only needs to read in the `abstract_*.txt.ready` files – the `alphas.txt` and `abstract_*.txt` files are provided for your reference only.

It is common with MCMC methods to discard the first  $X$  samples to avoid using samples that are highly correlated with the arbitrary starting assignment (this is called “burning in”). Use  $X = 50$  for your Gibbs sampling implementations.

For each of the abstracts,

- (a) Use your code to generate an accurate estimate of  $E[\theta]$  using collapsed Gibbs sampling with a high number of iterations (e.g.  $10^4$ ). Use this as ground truth.

The following formula can be used to obtain an estimate of  $\theta$  from the collapsed Gibbs sampler (where  $T$  is the number of samples):

$$E[\theta_i] = \frac{T\alpha_i + \sum_{t=1}^T \sum_{n=1}^N \mathbf{1}[z_n^t = i]}{T(\sum_{i=1}^k \alpha_i + N)}$$

- (b) Plot the  $\ell_2$  error on your estimate of  $E[\theta]$  as a function of the number of iterations for each of the algorithms.

**Only include in your solutions the plot for the data file** NIPS2008\_0517. The remaining files are provided for your own experimentation.

You may use the programming language of your choice. We recommend first checking that packages are available to (1) sample from a Dirichlet distribution, and (2) compute the Digamma function  $\Psi(x)$ , as these will simplify your coding. For example, see Python's `numpy.random.mtrand.dirichlet` and `scipy.special.psi`.

## References

- [1] David M. Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.