Probabilistic Graphical Models, Spring 2013

Problem Set 6: Monte-Carlo methods and variational inference

Due: Thursday, April 18, 2012 at 5pm (in class)

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, $m = 1, \ldots, M$
  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 α and β as fixed hyperparameters. Note that β is a matrix, with one column per topic, and the Multinomial variable z_{mn} selects one of the columns of β to yield multinomial probabilities for w_{mn}.

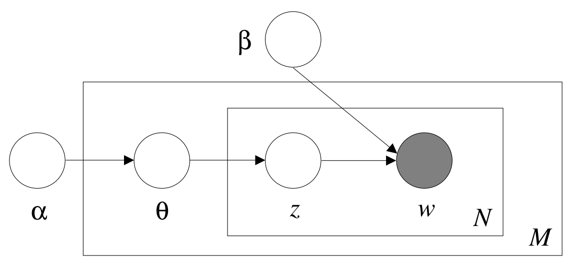

Figure 1: Graphical structure of the LDA model.

For Problems 1–3 below, hand in your full derivation, not just the final solution.

1. Derive a Gibbs sampler for the LDA model (i.e., write down the set of conditional probabilities for the sampler; see page 506 of Koller & Friedman).

   You may find it helpful to refer to your solutions from Problem Set 3.

2. 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 θ_m) and are now only sampling $\mathbf{z}$.

3. Derive a mean-field algorithm for inference in the LDA model by minimizing the KL-divergence $D(q_{\gamma_m,\phi_m}(\theta, \mathbf{z})||p(\theta, \mathbf{z} \mid \mathbf{w}))$ with respect to the variational parameters ϕ and γ, where $q_{\gamma,\phi}(\theta, \mathbf{z}) = q_\gamma(\theta) \prod_n q_{\phi_n}(z_n)$, $q_\gamma(\theta)$ is a Dirichlet, and the $q_{\phi_n}(z_n)$ are Multinomial. Feel free to refer to the LDA paper if you have difficulty [1]. In particular, you will probably want to use the fact that:

$$\mathrm{E}_{q_\gamma}[\log \theta_i] = \Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j)$$

4. Implement each of the three inference algorithms that you derived. You will then run your algorithms to find the posterior topic distribution θ for an input document.

   We have previously learned the parameters (i.e., α and β) 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, α_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,
     - α_i, for $i = 1, \ldots, k$,
     - for every word w_n, the word itself followed by $\beta_{w_n,i}$ for $i = 1, \ldots, k$.

   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 θ from the collapsed Gibbs sampler (where T is the number of samples):

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

   (b) Plot the ℓ_2 error on your estimate of $E[\theta]$ as a function of the number of iterations for each of the three algorithms.

   (c) Which algorithm converges fastest? Do all algorithms return an accurate estimate of $E[\theta_m]$ when run for a sufficiently long time? Explain your answers.

   Print and hand in only the plot for the data file `NIPS2008_0517`. The remaining files are provided for your own experimentation.

   Print all code and submit together with your solutions. As with the earlier problem sets, 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.