DATA20023 Bayesian Machine Learning

Exercise 2: MCMC and LDA **Due Feb 3**

1 MCMC for Poisson-gamma

Consider a model

$$\begin{aligned} x_n &\sim \text{Poisson}(uv), \\ u &\sim \text{Gamma}(5, 1), \\ v &\sim \text{Gamma}(4, 4). \end{aligned}$$

where individual observations x_n are conditionally independent of each other. Use MCMC to infer $p(u, v|\mathbf{x})$ for the set of observations $\mathbf{x} = [3, 4, 3, 9, 10, 3, 2, 3, 3, 2]$

(a) Write down the logarithm of the joint log-probability $\log p(\mathbf{x}, u, v)$

(b) Derive a Gibbs sampler for this model by providing the conditional distributions $p(u|v, \mathbf{x})$ and $p(v|u, \mathbf{x})$

(c) Implement both **Gibbs sampler** and **Metropolis-Hastings sampler** for this model (you can build on the examples in the Lecture 3 notebook, so this should be easy).

(d) For both methods, create plots showing marginal histograms of the parameters values and the sampling chains. Furthermore, provide mean and variance of the posterior distribution for both parameters.

(e) Explain briefly the differences between Metropolis-Hastings and Gibbs in this case. Which one would you use?

2 Latent Dirichlet allocation

2.1 Material

This exercise is about *latent Dirichlet allocation* (LDA) explained in Sections 27.1 and 27.3 of Murphy's book. Read the sections in detail. Pay attention in particular to:

- Plate diagram in Figure 27.2 (b)
- The Gibbs sampling in sub-section 27.3.4
- Footnote 1 on page 950 about notation – this exercise follows the notation of the book

2.2 Gibbs sampling

The data file has $N = 40$ 'text documents' with vocabulary of $V = 100$ words. To simplify coding, all documents are 50 words lomg and the 'words' numbers between 0 and 99. Hence you can store all documents in a single $N \times 50$ numerical matrix, rather than needing to create dictionaries or lists.

Complete the following tasks related to Gibbs sampling for this model:

1. Equations (27.30) - (27.32) provide conditional distributions for sampling π and q_{il} (the responsible topic for each word) and b_k (the topics)[1]. Implement the direct Gibbs sampler and run it on the data using $\alpha = 0.1$, $\gamma = 0.1$ and $K = 10$ topics. Does the result look good, corresponding to roughly what you would expect (e.g. placing words occurring in the same documents in the same topic etc)? Try to think of a good way to explore the results.
2. Equation (27.37) provides the probabilities for directly sampling q_{il} for one word conditional on the allocations for all others, integrating over π and b. Note that you can still use equations (27.31) and (27.32) to obtain π and b for interpretation. Implement and try also this algorithm.
3. Using one of the algorithms, briefly explore how the results change if you use clearly smaller or larger hyperparameters α and γ. Try also learning a solution with smaller or larger K.
4. LDA results are often interpreted by showing for each topic a set of likely words, selected based on one posterior sample (the last). However, this is not particularly Bayesian way of reporting results as it does not account for the uncertainty in any way. Try to think of a better way of characterising the topics and briefly describe how you would use multiple samples from the posterior. Note that you do not need to implement your idea, just describe it.

Hints:

- Start by checking dimensionalities of all terms, so that you definitely understand the model right.
- You will need code that computes the counts c_{ivk} based on q_{il}. Then you can implement both algorithms by summing over c_{ivk} along suitable dimensions. Pay attention to the indexing; in c_{ivk} the index v is the word identity (one of 'a', 'b' etc), which is **not** the same thing as l that gives the running index of each word. For instance, in y_3 the first word of the document would correspond to $l = 1$ but $v = 3$ (assuming 'a'=1, 'b'=2, etc).
- Collapsed Gibbs is probably easiest to implement by storing and updating both c_{ivk} and q_{il}. To allocate a new sample you first subtract one from c_{ivk} based on current value q_{il} before computing all the sums for equation (27.37). After sampling a new value you add one to the right element and update the corresponding element in q_{il}. This is inefficient compared to explicitly maintaining also the different marginal sums, but does not matter for our small data.
- You can initialize both methods by sampling π and b from their priors, and then use equation (27.30) once to get initial q.

[1] The book suddenly changes to x_{il} here instead of y_{il}, but that's a mistake.