Lecture 10: Gibbs Sampling in LDA

4F13: Machine Learning

Joaquin Quiñonero-Candela and Carl Edward Rasmussen

Department of Engineering
University of Cambridge

http://mlg.eng.cam.ac.uk/teaching/4f13/

Variants of the Metropolis algorithm

Instead of proposing a new state by changing simultaneously all components of the state, you can concatenate different proposals changing one component at a time.

For example, $q_j(\mathbf{x}'|\mathbf{x})$, such that $x_i' = x_i$, $\forall i \neq j$.

Now, cycle through the $q_j(\mathbf{x}'|\mathbf{x})$ in turn.

This is valid as

- each iteration obeys detailed balance
- the *sequence* guarantees ergodicity

Gibbs sampling

Updating one coordinate at a time, and choosing the proposal distribution to be the conditional distribution of that variable given all other variables $q(x_i'|\mathbf{x}) = p(x_i'|x_{\neq i})$ we get

$$a \;=\; \min\Big(1, \frac{p(x_{\neq i})p(x_i'|x_{\neq i})p(x_i|x_{\neq i}')}{p(x_{\neq i})p(x_i|x_{\neq i})p(x_i'|x_{\neq i})}\Big) \;=\; 1,$$

i.e., we get an algorithm which always accepts. This is called the Gibbs sampling algorithm.

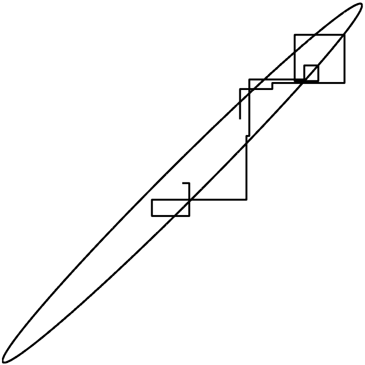
If you can compute (and sample from) the conditionals, you can apply Gibbs sampling.

This algorithm is completely parameter free.

Can also be applied to subsets of variables.

Example: Gibbs Sampling

20 iterations of Gibbs sampling on a bivariate Gaussian



Notice that strong correlations can slow down Gibbs sampling.

Collapsed Gibbs sampler for LDA

In the LDA model, we can integrate out the parameters of the multinomial distributions, θ_d and β, and just keep the latent counts z_{id}. This is called a *collapsed* Gibbs sampler.

Recall, that the predictive distribution for a symmetric Dirichlet is given by

$$p_i \;=\; \frac{\alpha + c_i}{\sum_j \alpha + c_j}.$$

Now, for Gibbs sampling, we need the predictive distribution for a single z_{id} given all other z_{id}, ie, given all the counts except for the count arrising from word i in document d.

The Gibbs update contains two parts, one from the topic distribution and one from the word distribution:

$$p(z_{id} = k) \;\propto\; \frac{\alpha + c^k_{-id}}{\sum_j \alpha + c^j_{-id}} \;\; \frac{\gamma + c^k_{-i}}{\sum_j \gamma + c^k_{-j}},$$

where c^k_{-id} is the counts of words from document d, excluding i being assigned to topic k and c^k_{-j} is the count of number of times word j was generated from topic k (again, excluding the current obsevation).

Per word Perplexity

In text modeling, performance is often given in terms of per word *perplexity*. The perplexity for a document is given by

$$\exp(-l/n),$$

where l is the joint log probability over the words in the document, and n is the number of words. Note, that the average is done in the log space.

A perplexity of g corresponds to the uncertainty associated with a die with g sides, which generates each new word.