Name : Tang Sophearin

Affiliation : Department of Intelligence Science and Technology, Nishida Laboratory

Student ID : 6930-30-3963

Latent Dirichlet Allocation: Parameter Estimation Using Gibbs Sampling

Latent Dirichlet Allocation

Latent Dirichlet Allocation is a generative probabilistic model used for topic modelling for documents with sets of topics. LDA generate topics for each document through the topic distribution and words from each topic through that distribution.

Each document consists of mixture topics and each topic is a probability distribution over words. Therefore, the probability of *i*th word in a given document having T topics is

$$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j), \quad [1]$$

where:

$P(w_i|z_i = j)$: probability of ith word generated from topics j;

$P(z_i = j)$: probablity of choosing a word from topics j in the current document.

LDA is different from a standard Bayesian classifier model. Instead of considering all words in a given document is generated from a single topic, LDA assumes words can be generated from multiple topics in a single document (soft classification).

There are two parameters to be estimated in LDA: ϕ and θ. Given *D* documents with *T* topics consisting of *W* words, ϕ is *T* multinomial distributions over *W* words and θ is *D* multinomial distributions over *T* topics. The main goal here is to find optimal ϕ distributions which gives high probability to the words appearing in some corpus. One might consider obtaining this estimate by maximizing $P(\mathrm{w}|\phi, \theta)$ applying Expectation-Maximization to equation 1. However, this technique is slow to converge and has local maxima problem. To address this problem, LDA makes assumption about the prior probability distribution of θ. Therefore, in LDA, to generate a new document, first we decide θ a Dirichlet distribution and pick some topics; then, words are generated from $\phi^{(j)}$ for each topic. Therefore, we need to maximize $P(w|\phi, \alpha) = \int P(w|\phi, \theta)P(\theta|\alpha)\, d\theta$, where $P(\theta)$ is a Dirichlet(α) distribution.

Gibbs Sampling

Instead of estimating ϕ and θ explicitly, Gibbs sampling method uses Monte Carlos procedure to find posterior distribution $P(\mathrm{z}|w)$; then, by examining this posterior, we can obtain ϕ and θ.

Probability model is:

$w_i | z_i, \phi^{(zi)}$ ~ Discrete($\phi^{(zi)}$)

ϕ ~ Dirichlet(β)

$z_i | \theta^{(di)}$ ~ Discrete($\theta^{(di)}$)

θ ~ Dirichlet(α)

Here, α and β are hyperparameters. The α controls the mixture of topics for any given document. Turn it down for less of a mixture of topics; turn it up for more mixture. The β hyperparameter controls the distribution of words per topic. Turn it down and the topics will have less words; turn it up and the topics will have more words.

Unfortunate, the following posterior distribution cannot be calculated in closed form solution because the denominators does not factorize and involves T^n terms (n is number of words).

$$P(\mathbf{z}|\mathbf{w}) = \frac{P(\mathbf{w}, \mathbf{z})}{\sum_{\mathbf{z}} P(\mathbf{w}, \mathbf{z})}.$$

To deal with this problem, we use Markov chain Monte Carlo to sample (Gibbs sampling). Target distribution is obtained by iterating through,

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha},$$

where $n_{-i,j}^{(\cdot)}$ is a word count not including z_i. The first part represents the probability of words under each topic, and the second part represents the probability of topics under each document. *n* are number of word counts under topics/documents; *W* is number of unique instances for words; *T* is number topics. For each iteration, we use state from previous iteration and sample z_i to obtain another new state. Finally, the posterior distribution converges to desired target distribution.

ϕ and θ can then be computed by

$$\hat{\phi}_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(\cdot)} + W\beta}$$

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_\cdot^{(d)} + T\alpha}.$$

Gibbs sampling in LDA algorithm performs quite well against other parameter estimation algorithms such as variational Bayes and expectation propagation while it is also computationally efficient because it only caches the number of times a word assigned to a topic and the number of times a topic occurs in a document for each iteration of Gibbs sampling.