

Topic Models

Charles Elkan

elkan@cs.ucsd.edu

November 20, 2008

Suppose that we have a collection of documents, and we want to find an organization for these, i.e. we want to do unsupervised learning. One approach is to train a mixture distribution to fit the data. This model is based on the assumption that the documents can be grouped into clusters. Formally, a mixture distribution is a probability density function of the form

$$f(x) = \sum_{k=1}^K \alpha_k f(x; \theta_k).$$

Here, K is the number of components in the mixture model. For each k , $f(x; \theta_k)$ is the pdf of component number k . The scalar α_k is the proportion of component number k .

The specific topic model we consider is called latent Dirichlet allocation (LDA). (The same abbreviation is also used for linear discriminant analysis, which is unrelated.) LDA is based on the intuition that each document contains words from multiple topics; the proportion of each topic in each document is different, but the topics themselves are the same for all documents.

1 Mathematical preliminaries

Before we can see the LDA model in mathematical detail, we need to review the Dirichlet distribution. Let γ be a vector of length S such that $\gamma_s \geq 0$ for all s and $\sum_{s=1}^S \gamma_s = 1$. The Dirichlet is a probability density function over all such vectors. The Dirichlet is useful because it is a well-defined prior distribution for

a multinomial: the set of all vectors γ satisfying the conditions above is precisely the set of all possible multinomial parameter vectors.

A Dirichlet distribution has parameter vector α of length S . Note that $\alpha_s > 0$ is required for all s , but there is no constraint on $\sum_{s=1}^S \alpha_s$. Concretely, the equation for the Dirichlet distribution is

$$p(\gamma|\alpha) = \frac{1}{D(\alpha)} \prod_{s=1}^S \gamma_s^{\alpha_s-1}$$

where the function D is defined as

$$D(\alpha) = \frac{\prod_{s=1}^S \Gamma(\alpha_s)}{\Gamma(\sum_{s=1}^S \alpha_s)}.$$

Here the Γ function is the standard continuous generalization of the factorial such that $\Gamma(k) = (k - 1)!$ if k is an integer.

The D function has a very important property that will be used below:

$$D(\alpha) = \int_{\gamma} \prod_i \gamma_i^{\alpha_i-1}$$

where the integral is over the set of all unit vectors γ .

2 The LDA model and Gibbs sampling

The generative process assumed by the LDA model is as follows:

- Given: Dirichlet distribution with parameter vector α of length K
- Given: Dirichlet distribution with parameter vector β of length V
- for topic number 1 to topic number K
 - draw a multinomial with parameter vector ϕ_k according to β
- for document number 1 to document number M
 - draw a topic distribution, i.e. a multinomial θ according to α
 - for each word in the document
 - draw a topic z according to θ
 - draw a word w according to ϕ_z

Note that z is an integer between 1 and K for each word.

For learning, the training data are the words in all documents. The prior distributions α and β are assumed to be fixed and known, as are the number K of topics, the number M of documents, the length N_m of each document, and the cardinality V of the vocabulary (i.e. the dictionary of all words). Learning has two goals: (i) to infer the document-specific multinomial θ for each document, and (ii) to infer the topic distribution ϕ_k for each topic.

The algorithm that we use for learning is called collapsed Gibbs sampling. It does not infer the θ and ϕ_k distributions directly. Instead, it infers the hidden values z for all words w .

Suppose that we know the value of z for every word in the corpus except word number i . The idea of Gibbs sampling is to draw a z value for word i randomly according to its distribution, then assume that we know this value, then draw a z value for another word, and so on. It can be proved that eventually this process converges to a correct distribution of z values for all words in the corpus. Note that Gibbs sampling never converges to a fixed z for each w ; instead it converges to a distribution of z values for each w .

Let \bar{w} be the sequence of words making up the entire corpus, and let \bar{z} be a corresponding sequence of z values. Note that \bar{w} is not a vector of word counts. Use the notation \bar{w}' to mean \bar{w} with word number i removed, so $\bar{w} = \{w_i, \bar{w}'\}$. Similarly, write $\bar{z} = \{z_i, \bar{z}'\}$. In order to do Gibbs sampling, we need to compute

$$p(z_i | \bar{z}', \bar{w}) = \frac{p(\bar{z}, \bar{w})}{p(\bar{z}', \bar{w})} = \frac{p(\bar{w} | \bar{z}) p(\bar{z})}{p(w_i | \bar{z}') p(\bar{w}' | \bar{z}') p(\bar{z}')}$$

for $z_i = 1$ to $z_i = K$. In principle the entire denominator can be ignored, because it is a constant that does not depend on z_i . However, we will pay attention to the second and third factors in the denominator, because they lead to cancellations with the numerator. So we will evaluate

$$p(z_i | \bar{z}', \bar{w}) \propto \frac{p(\bar{w} | \bar{z}) p(\bar{z})}{p(\bar{w}' | \bar{z}') p(\bar{z}')} \tag{1}$$

3 Mathematical details

We will work out the four factors of (1) one by one. Consider $p(\bar{z})$ first, and let \bar{z} refer temporarily to just document number m . For this document,

$$p(\bar{z} | \theta) = \prod_{i=1}^{N_m} p(z_i) = \prod_{k=1}^K \theta_k^{n_{mk}}$$

where n_{mk} is the number of times $z_i = k$ within document m , and θ is the multinomial parameter vector for document m . What we really want is $p(\bar{z}|\alpha)$, which requires averaging over all possible θ . This is

$$p(\bar{z}|\alpha) = \int_{\theta} p(\theta|\alpha)p(\bar{z}|\theta).$$

By the definition of the Dirichlet distribution,

$$p(\theta|\alpha) = \frac{\prod_{k=1}^K \theta_k^{\alpha_k - 1}}{D(\alpha)}.$$

Therefore,

$$\begin{aligned} p(\bar{z}|\alpha) &= \frac{1}{D(\alpha)} \int_{\theta} \prod_{k=1}^K \theta_k^{\alpha_k - 1} \prod_{k=1}^K \theta_k^{n_{mk}} \\ &= \frac{1}{D(\alpha)} \int_{\theta} \prod_{k=1}^K \theta_k^{n_{mk} + \alpha_k - 1} \\ &= \frac{D(\bar{n}_m + \alpha)}{D(\alpha)} \end{aligned}$$

where \bar{n}_m is the vector of topic counts $\langle n_{m1}, \dots, n_{mK} \rangle$ for document number m . Similarly,

$$p(\bar{z}'|\alpha) = \frac{D(\bar{n}'_m + \alpha)}{D(\alpha)}$$

where \bar{n}'_m is \bar{n}_m with one subtracted from the count for topic number z_i . For the corpus of all M documents,

$$p(\bar{z}|\alpha) = \prod_{m=1}^M \frac{D(\bar{n}_m + \alpha)}{D(\alpha)} \quad (2)$$

and

$$p(\bar{z}'|\alpha) = \prod_{m=1}^M \frac{D(\bar{n}'_m + \alpha)}{D(\alpha)}. \quad (3)$$

Note that all factors except one are identical in (2) and (3).

Referring back to (1), consider $p(\bar{w}|\bar{z})$, which means $p(\bar{w}|\bar{z}, \beta)$. This is

$$\int_{\Phi} p(\Phi|\beta)p(\bar{w}|\bar{z}, \Phi)$$

where Φ is a collection of K different topic distributions ϕ_k . Again by the definition of the Dirichlet distribution,

$$p(\Phi|\beta) = \prod_{k=1}^K \frac{1}{D(\beta)} \prod_{t=1}^V \phi_{kt}^{\beta_t-1}.$$

Now consider $p(\bar{w}|\bar{z}, \Phi)$. To evaluate this, group the words w_i together according to which topic z_i they come from:

$$p(\bar{w}|\bar{z}, \Phi) = \prod_{k=1}^K \prod_{i:z_i=k} p(w_i|z_i, \Phi) = \prod_{k=1}^K \prod_{t=1}^V \phi_{kt}^{q_{kt}}$$

where q_{kt} is the number of times that word t occurs with topic k in the whole corpus. We get that

$$\begin{aligned} p(\bar{w}|\bar{z}, \beta) &= \int_{\Phi} \left(\prod_{k=1}^K \frac{1}{D(\beta)} \prod_{t=1}^V \phi_{kt}^{\beta_t-1} \right) \left(\prod_{k=1}^K \prod_{t=1}^V \phi_{kt}^{q_{kt}} \right) \\ &= \int_{\Phi} \prod_{k=1}^K \frac{1}{D(\beta)} \prod_{t=1}^V \phi_{kt}^{\beta_t-1+q_{kt}} \\ &= \prod_{k=1}^K \int_{\phi_k} \frac{1}{D(\beta)} \prod_{t=1}^V \phi_{kt}^{\beta_t-1+q_{kt}} \\ &= \prod_{k=1}^K \frac{1}{D(\beta)} D(\bar{q}_k + \beta) \end{aligned}$$

where \bar{q}_k is the vector of counts over the whole corpus of words belonging to topic k . This equation is similar to (2), with the corpus divided into K topics instead of into M documents.

Referring back again to (1), we get that $p(z_i|\bar{z}', \bar{w})$ is proportional to

$$\prod_{k=1}^K \frac{D(\bar{q}_k + \beta)}{D(\beta)} \prod_{k=1}^K \frac{D(\beta)}{D(\bar{q}'_k + \beta)} \prod_{m=1}^M \frac{D(\bar{n}_m + \alpha)}{D(\alpha)} \prod_{m=1}^M \frac{D(\alpha)}{D(\bar{n}'_m + \alpha)}.$$

The $D(\beta)$ and $D(\alpha)$ factors obviously cancel above. The products can be eliminated also because $\bar{q}_k + \beta = \bar{q}'_k + \beta$ except when the topic $k = z_i$, and $\bar{n}_m + \alpha = \bar{n}'_m + \alpha$ except for the document m that word i belongs to. So,

$$p(z_i|\bar{z}', \bar{w}) \propto D(\bar{q}_{z_i} + \beta) \frac{1}{D(\bar{q}'_{z_i} + \beta)} D(\bar{n}_m + \alpha) \frac{1}{D(\bar{n}'_m + \alpha)}$$

For any vector γ , the definition of the D function is

$$D(\gamma) = \frac{\prod_s \Gamma(\gamma_s)}{\Gamma(\sum_s \gamma_s)}$$

where s indexes the components of γ . Using this definition, $p(z_i = j|\bar{z}', \bar{w})$ is proportional to

$$\frac{\prod_t \Gamma(q_{jt} + \beta_t) \Gamma(\sum_t q'_{jt} + \beta_t) \prod_k \Gamma(n_{mk} + \alpha_k) \Gamma(\sum_k n'_{mk} + \alpha_k)}{\Gamma(\sum_t q_{jt} + \beta_t) \prod_t \Gamma(q'_{jt} + \beta_t) \Gamma(\sum_k n_{mk} + \alpha_k) \prod_k \Gamma(n'_{mk} + \alpha_k)}.$$

Now $q_{jt} + \beta_t = q'_{jt} + \beta_t$ except when $t = w_i$, in which case $q_{jt} + \beta_t = q'_{jt} + \beta_t + 1$, so

$$\frac{\prod_t \Gamma(q_{jt} + \beta_t)}{\prod_t \Gamma(q'_{jt} + \beta_t)} = \frac{\Gamma(q'_{jw_i} + \beta_{w_i} + 1)}{\Gamma(q'_{jw_i} + \beta_{w_i})}.$$

Applying the fact that $\Gamma(x + 1)/\Gamma(x) = x$ yields

$$\frac{\Gamma(q'_{jw_i} + \beta_{w_i} + 1)}{\Gamma(q'_{jw_i} + \beta_{w_i})} = q'_{jw_i} + \beta_{w_i}.$$

Similarly,

$$\frac{\prod_k \Gamma(n_{mk} + \alpha_k)}{\prod_k \Gamma(n'_{mk} + \alpha_k)} = n'_{mj} + \alpha_j$$

where $j = z_i$ is the candidate topic assignment of word i .

Summing over all words t in the vocabulary gives

$$\sum_{t=1}^V q_{jt} + \beta_t = 1 + \sum_{t=1}^V q'_{jt} + \beta_t.$$

so

$$\frac{\Gamma(\sum_t q'_{jt} + \beta_t)}{\Gamma(\sum_t q_{jt} + \beta_t)} = \frac{1}{\sum_t q'_{jt} + \beta_t}$$

and similarly

$$\frac{\Gamma(\sum_k n'_{mk} + \alpha_k)}{\Gamma(\sum_k n_{mk} + \alpha_k)} = \frac{1}{\sum_k n'_{mk} + \alpha_k}.$$

Putting the simplifications above together, $p(z_i = j | \bar{z}', \bar{w})$ is proportional to

$$\frac{q'_{jw_i} + \beta_{w_i}}{\sum_t q'_{jt} + \beta_t} \frac{n'_{mj} + \alpha_j}{\sum_k n'_{mk} + \alpha_k}. \quad (4)$$

This result says that occurrence number i is more likely to belong to topic j if q'_{jw_i} and/or n'_{mj} are large, i.e. if the same word occurs often with topic j elsewhere in the corpus, and/or if topic j occurs often elsewhere inside document m .

4 Implementation notes

In expression (4) the vectors α and β are fixed and known. Each value q'_{jw_i} depends on the current assignment of topics to each appearance of the word w_i throughout the corpus, not including the appearance as word number i . These assignments can be initialized in any desired way, and are then updated one at a time by the Gibbs sampling process. Each value n'_{mj} is the count of how many words within document m are assigned to topic j , not including word number i . These counts can be computed easily after the initial topic assignments are chosen, and then they can be updated in constant time whenever the topic assigned to a word changes.

The LDA generative model treats each word in each document individually. However, the specific order in which words are generated does not influence the probability of a document according to the generative model. Similarly, the Gibbs sampling algorithm works with a specific ordering of the words in the corpus. However, any ordering will do. Hence, the sequence of words inside each training document does not need to be known. The only training information needed for each document is how many times each word appears in it. Therefore, the LDA model can be learned from the standard bag-of-words representation of documents.

The standard approach to implementing Gibbs sampling iterates over every word of every document, taking the words in some arbitrary order. For each word, expression (4) is evaluated for each alternative topic j . For each j , evaluating (4) requires constant time, so the time to perform one epoch of Gibbs sampling is $O(NK)$ where N is the total number of words in all documents and K is the number of topics.