WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Languages
Deutsch
Español
Français
한국어
Português
Русский
中文
5 more
Edit links

Print/export
Download as PDF
Printable version

Latent Dirichlet allocation

From Wikipedia, the free encyclopedia

Not to be confused with linear discriminant analysis.

This article **may be too technical for most readers to understand**. Please help improve it to make it understandable to non-experts, without removing the technical details. *(August 2017) (Learn how and when to remove this template message)*

In natural language processing, the **latent Dirichlet allocation** (**LDA**) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. LDA is an example of a topic model and belongs to the machine learning toolbox and in wider sense to the artificial intelligence toolbox.

Contents [hide]

History [edit]

In the context of population genetics, LDA was proposed by J. K. Pritchard, M. Stephens and P. Donnelly in 2000.[1][2]

LDA was applied in machine learning by David Blei, Andrew Ng and Michael I. Jordan in 2003.[3]

Overview [edit]

Evolutionary biology and bio-medicine [edit]

In evolutionary biology and bio-medicine, the model is used to detect the presence of structured genetic variation in a group of individuals. The model assumes that alleles carried by individuals under study have origin in various extant or past populations. The model and various inference algorithms allow scientist to estimate the allele frequencies in those source populations and the origin of alleles carried by individuals under study. The source populations can be interpreted ex-post in terms of various evolutionary scenarios. In association studies, detecting the presence of genetic structure is considered a necessary preliminary step to avoid confounding.

Engineering [edit]

One example of LDA in engineering is to automatically classify documents and estimate their relevance to various topics.

In LDA, each document may be viewed as a mixture of various topics where each document is considered to have a set of topics that are assigned to it via LDA. This is identical to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution is assumed to have a sparse Dirichlet prior. The sparse Dirichlet priors encode the intuition that documents cover only a small set of topics and that topics use only a small set of words frequently. In practice, this results in a better disambiguation of words and a more precise assignment of documents to topics. LDA is a generalization of the pLSA model, which is equivalent to LDA under a uniform Dirichlet prior distribution.[4]

For example, an LDA model might have topics that can be classified as **CAT_related** and **DOG_related**. A topic has probabilities of generating various words, such as *milk*, *meow*, and *kitten*, which can be classified and interpreted by the viewer as "CAT_related". Naturally, the word *cat* itself will have high probability given this topic. The **DOG_related** topic likewise has probabilities of generating each word: *puppy*, *bark*, and *bone* might have high probability. Words without special relevance, such as *"the"* (see function word), will have roughly even probability between classes (or can be placed into a separate category). A topic is neither semantically nor epistemologically strongly defined. It is identified on the basis of automatic detection of the likelihood of term co-occurrence. A lexical word may occur in several topics with a different probability, however, with a different typical set of neighboring words in each topic.

Each document is assumed to be characterized by a particular set of topics. This is similar to the standard bag of words model assumption, and makes the individual words exchangeable.

Model [edit]

With plate notation, which is often used to represent probabilistic graphical models (PGMs), the dependencies among the many variables can be captured concisely. The boxes are "plates" representing replicates, which are repeated entities. The outer plate represents documents, while the inner plate represents the repeated word positions in a given document; each position is associated with a choice of topic and word. The variable names are defined as follows:

M denotes the number of documents
N is number of words in a given document (document *i* has N_i words)
α is the parameter of the Dirichlet prior on the per-document topic distributions
β is the parameter of the Dirichlet prior on the per-topic word distribution
θ_i is the topic distribution for document *i*
φ_k is the word distribution for topic *k*
z_{ij} is the topic for the *j*-th word in document *i*
w_{ij} is the specific word.

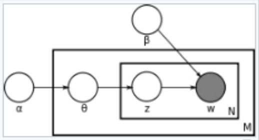

Plate notation representing the LDA model.

The fact that W is grayed out means that words w_{ij} are the only observable variables, and the other variables are latent variables. As proposed in the original paper[3], a sparse Dirichlet prior can be used to model the topic-word distribution, following the intuition that the probability distribution over words in a topic is skewed, so that only a small set of words have high probability. The resulting model is the most widely applied variant of LDA today. The plate notation for this model is shown on the right, where K denotes the number of topics and $\varphi_1, \dots, \varphi_K$ *are* V-dimensional vectors storing the parameters of the Dirichlet-distributed topic-word distributions (V is the number of words in the vocabulary).

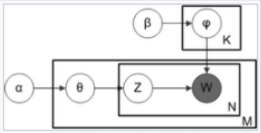

Plate notation for LDA with Dirichlet-distributed topic-word distributions

It is helpful to think of the entities represented by θ and φ as matrices created by decomposing the original document-word matrix that represents the corpus of documents being modeled. In this view, θ consists of rows defined by documents and columns defined by topics, while φ consists of rows defined by topics and columns defined by words. Thus, $\varphi_1, \dots, \varphi_K$ refers to a set of rows, or vectors, each of which is a distribution over words, and $\theta_1, \dots, \theta_M$ refers to a set of rows, each of which is a distribution over topics.

Generative process [edit]

To actually infer the topics in a corpus, we imagine a generative process whereby the documents are created, so that we may infer, or reverse engineer, it. We imagine the generative process as follows. Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over all the words. LDA assumes the following generative process for a corpus D consisting of M documents each of length N_i:

1. Choose $\theta_i \sim \operatorname{Dir}(\alpha)$, where $i \in \{1, \dots, M\}$ and $\operatorname{Dir}(\alpha)$ is a Dirichlet distribution with a symmetric parameter α which typically is sparse ($\alpha < 1$)

2. Choose $\varphi_k \sim \operatorname{Dir}(\beta)$, where $k \in \{1, \dots, K\}$ and β typically is sparse

3. For each of the word positions i, j, where $i \in \{1, \dots, M\}$, and $j \in \{1, \dots, N_i\}$

(a) Choose a topic $z_{i,j} \sim \operatorname{Multinomial}(\theta_i)$.

(b) Choose a word $w_{i,j} \sim \operatorname{Multinomial}(\varphi_{z_{i,j}})$.

(Note that *multinomial distribution* here refers to the multinomial with only one trial, which is also known as the categorical distribution.)

The lengths N_i are treated as independent of all the other data generating variables (w and z). The subscript is often dropped, as in the plate diagrams shown here.

Definition [edit]

A formal description of LDA is as follows:

Definition of variables in the model

Variable	Type	Meaning
K	integer	number of topics (e.g. 50)
V	integer	number of words in the vocabulary (e.g. 50,000 or 1,000,000)
M	integer	number of documents
$N_{d=1\dots M}$	integer	number of words in document *d*
N	integer	total number of words in all documents; sum of all N_d values, i.e. $N = \sum_{d=1}^{M} N_d$
$\alpha_{k=1\dots K}$	positive real	prior weight of topic *k* in a document; usually the same for all topics; normally a number less than 1, e.g. 0.1, to prefer sparse topic distributions, i.e. few topics per document
$\boldsymbol{\alpha}$	*K*-dimensional vector of positive reals	collection of all α_k values, viewed as a single vector
$\beta_{w=1\dots V}$	positive real	prior weight of word *w* in a topic; usually the same for all words; normally a number much less than 1, e.g. 0.001, to strongly prefer sparse word distributions, i.e. few words per topic
$\boldsymbol{\beta}$	*V*-dimensional vector of positive reals	collection of all β_w values, viewed as a single vector
$\varphi_{k=1\dots K,w=1\dots V}$	probability (real number between 0 and 1)	probability of word *w* occurring in topic *k*
$\boldsymbol{\varphi}_{k=1\dots K}$	*V*-dimensional vector of probabilities, which must sum to 1	distribution of words in topic *k*
$\theta_{d=1\dots M,k=1\dots K}$	probability (real number between 0 and 1)	probability of topic *k* occurring in document *d*
$\boldsymbol{\theta}_{d=1\dots M}$	*K*-dimensional vector of probabilities, which must sum to 1	distribution of topics in document *d*
$z_{d=1\dots M,w=1\dots N_d}$	integer between 1 and *K*	identity of topic of word *w* in document *d*
$\mathbf{Z}$	*N*-dimensional vector of integers between 1 and *K*	identity of topic of all words in all documents
$w_{d=1\dots M,w=1\dots N_d}$	integer between 1 and *V*	identity of word *w* in document *d*
$\mathbf{W}$	*N*-dimensional vector of integers between 1 and *V*	identity of all words in all documents

We can then mathematically describe the random variables as follows:

$$
\begin{aligned}
\boldsymbol{\varphi}_{k=1\dots K} &\sim \operatorname{Dirichlet}_V(\boldsymbol{\beta}) \\
\boldsymbol{\theta}_{d=1\dots M} &\sim \operatorname{Dirichlet}_K(\boldsymbol{\alpha}) \\
z_{d=1\dots M,w=1\dots N_d} &\sim \operatorname{Categorical}_K(\boldsymbol{\theta}_d) \\
w_{d=1\dots M,w=1\dots N_d} &\sim \operatorname{Categorical}_V(\boldsymbol{\varphi}_{z_{dw}})
\end{aligned}
$$

Inference [edit]

See also: Dirichlet-multinomial distribution

Learning the various distributions (the set of topics, their associated word probabilities, the topic of each word, and the particular topic mixture of each document) is a problem of statistical inference.

Monte Carlo simulation [edit]

The original paper by Pritchard et al.[1] used approximation of the posterior distribution by Monte Carlo simulation. Alternative proposal of inference techniques include Gibbs sampling.[5]

Variational Bayes [edit]

The original ML paper used a variational Bayes approximation of the posterior distribution;[3]

Likelihood maximization [edit]

A direct optimization of the likelihood with a block relaxation algorithm proves to a fast alternative to MCMC.[6]

Unknown number of populations/topics [edit]

In practice, the most adequate number of populations or topics is not known beforehand. It can be estimated by estimation of the posterior distribution with [Reversible jump Markov chain Monte Carlo][7]

Alternative approaches [edit]

Alternative approaches include expectation propagation.[8]

Recent research has been focused on speeding up the inference of latent Dirichlet Allocation to support the capture of a massive number of topics in a large number of documents. The update equation of the collapsed Gibbs sampler mentioned in the earlier section has a natural sparsity within it that can be taken advantage of. Intuitively, since each document only contains a subset of topics K_d, and a word also only appears in a subset of topics K_w, the above update equation could be rewritten to take advantage of this sparsity.[9]

$$p(Z_{d,n}=k) \propto \frac{\alpha\beta}{C_k^{\neg n}+V\beta} + \frac{C_k^d\beta}{C_k^{\neg n}+V\beta} + \frac{C_k^w(\alpha+C_k^d)}{C_k^{\neg n}+V\beta}$$

In this equation, we have three terms, out of which two of them are sparse, and the other is small. We call these terms a, b and c respectively. Now, if we normalize each term by summing over all the topics, we get:

$$A = \sum_{k=1}^{K} \frac{\alpha\beta}{C_k^{\neg n}+V\beta}$$

$$B = \sum^{K} \frac{C_k^d\beta}{C_k^{\neg n}+V\beta}$$

$$C = \sum_{k=1}^{K} \frac{C_k^w(\alpha + C_k^d)}{C_k^{\neg n} + V\beta}$$

Here, we can see that B is a summation of the topics that appear in document d, and C is also a sparse summation of the topics that a word w is assigned to across the whole corpus. A on the other hand, is dense but because of the small values of α & β, the value is very small compared to the two other terms.

Now, while sampling a topic, if we sample a random variable uniformly from $s \sim U(s| \mid A + B + C)$, we can check which bucket our sample lands in. Since A is small, we are very unlikely to fall into this bucket; however, if we do fall into this bucket, sampling a topic takes $O(K)$ time (same as the original Collapsed Gibbs Sampler). However, if we fall into the other two buckets, we only need to check a subset of topics if we keep a record of the sparse topics. A topic can be sampled from the B bucket in $O(K_d)$ time, and a topic can be sampled from the C bucket in $O(K_w)$ time where K_d and K_w denotes the number of topics assigned to the current document and current word type respectively.

Notice that after sampling each topic, updating these buckets is all basic $O(1)$ arithmetic operations.

Aspects of computational details [edit]

Following is the derivation of the equations for collapsed Gibbs sampling, which means φs and θs will be integrated out. For simplicity, in this derivation the documents are all assumed to have the same length N. The derivation is equally valid if the document lengths vary.

According to the model, the total probability of the model is:

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) P(W_{j,t} \mid \varphi_{Z_{j,t}}),$$

where the bold-font variables denote the vector version of the variables. First, $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$ need to be integrated out.

$$\begin{aligned} P(\boldsymbol{Z}, \boldsymbol{W}; \alpha, \beta) &= \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\varphi}} P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) \, d\boldsymbol{\varphi} \, d\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\varphi}} \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} \prod_{t=1}^{N} P(W_{j,t} \mid \varphi_{Z_{j,t}}) \, d\boldsymbol{\varphi} \int_{\boldsymbol{\theta}} \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) \, d\boldsymbol{\theta}. \end{aligned}$$

All the θs are independent to each other and the same to all the φs. So we can treat each θ and each φ separately. We now focus only on the θ part.

$$\int_{\boldsymbol{\theta}} \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) \, d\boldsymbol{\theta} = \prod_{j=1}^{M} \int_{\theta_j} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) \, d\theta_j.$$

We can further focus on only one θ as the following:

$$\int_{\theta_j} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) \, d\theta_j.$$

Actually, it is the hidden part of the model for the j^{th} document. Now we replace the probabilities in the above equation by the true distribution expression to write out the explicit equation.

$$\int_{\theta_j} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) \, d\theta_j = \int_{\theta_j} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_{j,i}^{\alpha_i - 1} \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) \, d\theta_j.$$

Let $n_{j,r}^i$ be the number of word tokens in the j^{th} document with the same word symbol (the r^{th} word in the vocabulary) assigned to the i^{th} topic. So, $n_{j,r}^i$ is three dimensional. If any of the three dimensions is not limited to a specific value, we use a parenthesized point $(\cdot)$ to denote. For example, $n_{j,(\cdot)}^i$ denotes the number of word tokens in the j^{th} document assigned to the i^{th} topic. Thus, the right most part of the above equation can be rewritten as:

$$\prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) = \prod_{i=1}^{K} \theta_{j,i}^{n_{j,(\cdot)}^i}.$$

So the θ_j integration formula can be changed to:

$$\int_{\theta_j} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_{j,i}^{\alpha_i - 1} \prod_{i=1}^{K} \theta_{j,i}^{n_{j,(\cdot)}^i} \, d\theta_j = \int_{\theta_j} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_{j,i}^{n_{j,(\cdot)}^i + \alpha_i - 1} \, d\theta_j.$$

Clearly, the equation inside the integration has the same form as the Dirichlet distribution. According to the Dirichlet distribution,

$$\int_{\theta_j} \frac{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^i + \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^i + \alpha_i)} \prod_{i=1}^{K} \theta_{j,i}^{n_{j,(\cdot)}^i + \alpha_i - 1} \, d\theta_j = 1.$$

Thus,

$$\begin{aligned} \int_{\theta_j} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) \, d\theta_j &= \int_{\theta_j} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_{j,i}^{n_{j,(\cdot)}^i + \alpha_i - 1} \, d\theta_j \\ &= \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^i + \alpha_i)}{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^i + \alpha_i\right)} \int_{\theta_j} \frac{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^i + \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^i + \alpha_i)} \prod_{i=1}^{K} \theta_{j,i}^{n_{j,(\cdot)}^i + \alpha_i - 1} \, d\theta_j \\ &= \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^i + \alpha_i)}{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^i + \alpha_i\right)}. \end{aligned}$$

Now we turn our attention to the $\boldsymbol{\varphi}$ part. Actually, the derivation of the $\boldsymbol{\varphi}$ part is very similar to the $\boldsymbol{\theta}$ part. Here we only list the steps of the derivation:

$$\begin{aligned} &\int_{\boldsymbol{\varphi}} \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} \prod_{t=1}^{N} P(W_{j,t} \mid \varphi_{Z_{j,t}}) \, d\boldsymbol{\varphi} \\ &= \prod_{i=1}^{K} \int_{\varphi_i} P(\varphi_i; \beta) \prod_{j=1}^{M} \prod_{t=1}^{N} P(W_{j,t} \mid \varphi_{Z_{j,t}}) \, d\varphi_i \\ &= \prod_{i=1}^{K} \int_{\varphi_i} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \prod_{r=1}^{V} \varphi_{i,r}^{\beta_r - 1} \prod_{r=1}^{V} \varphi_{i,r}^{n_{(\cdot),r}^i} \, d\varphi_i \\ &= \prod_{i=1}^{K} \int_{\varphi_i} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \prod_{r=1}^{V} \varphi_{i,r}^{n_{(\cdot),r}^i + \beta_r - 1} \, d\varphi_i \\ &= \prod_{i=1}^{K} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \frac{\prod_{r=1}^{V} \Gamma(n_{(\cdot),r}^i + \beta_r)}{\Gamma\left(\sum_{r=1}^{V} n_{(\cdot),r}^i + \beta_r\right)}. \end{aligned}$$

For clarity, here we write down the final equation with both $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ integrated out:

$$P(\boldsymbol{Z}, \boldsymbol{W}; \alpha, \beta) = \prod_{j=1}^{M} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^i + \alpha_i)}{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^i + \alpha_i\right)} \times \prod_{i=1}^{K} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \frac{\prod_{r=1}^{V} \Gamma(n_{(\cdot),r}^i + \beta_r)}{\Gamma\left(\sum_{r=1}^{V} n_{(\cdot),r}^i + \beta_r\right)}.$$

$$\prod_{j=1} \prod_{i=1}\Gamma(\alpha_i) \quad \Gamma\left(\sum_{i=1} n^i_{j,(\cdot)}+\alpha_i\right) \qquad \prod_{i=1} \prod_{r=1}\Gamma(\beta_r) \quad \Gamma\left(\sum_{r=1} n^i_{(\cdot),r}+\beta_r\right)$$

The goal of Gibbs Sampling here is to approximate the distribution of $P(\boldsymbol{Z} \mid \boldsymbol{W};\alpha,\beta)$. Since $P(\boldsymbol{W};\alpha,\beta)$ is invariable for any of Z, Gibbs Sampling equations can be derived from $P(\boldsymbol{Z},\boldsymbol{W};\alpha,\beta)$ directly. The key point is to derive the following conditional probability:

$$P(Z_{(m,n)} \mid \boldsymbol{Z_{-(m,n)}},\boldsymbol{W};\alpha,\beta)=\frac{P(Z_{(m,n)},\boldsymbol{Z_{-(m,n)}},\boldsymbol{W};\alpha,\beta)}{P(\boldsymbol{Z_{-(m,n)}},\boldsymbol{W};\alpha,\beta)},$$

where $Z_{(m,n)}$ denotes the Z hidden variable of the n^{th} word token in the m^{th} document. And further we assume that the word symbol of it is the v^{th} word in the vocabulary. $\boldsymbol{Z_{-(m,n)}}$ denotes all the Zs but $Z_{(m,n)}$. Note that Gibbs Sampling needs only to sample a value for $Z_{(m,n)}$, according to the above probability, we do not need the exact value of

$$P\left(Z_{m,n} \mid \boldsymbol{Z_{-(m,n)}},\boldsymbol{W};\alpha,\beta\right)$$

but the ratios among the probabilities that $Z_{(m,n)}$ can take value. So, the above equation can be simplified as:

$$
\begin{aligned}
&P(Z_{(m,n)}=v \mid \boldsymbol{Z_{-(m,n)}},\boldsymbol{W};\alpha,\beta)\\
&\propto P(Z_{(m,n)}=v,\boldsymbol{Z_{-(m,n)}},\boldsymbol{W};\alpha,\beta)\\
&=\left(\frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\right)^{M}\prod_{j\neq m}\frac{\prod_{i=1}^{K}\Gamma\left(n^i_{j,(\cdot)}+\alpha_i\right)}{\Gamma\left(\sum_{i=1}^{K}n^i_{j,(\cdot)}+\alpha_i\right)}\left(\frac{\Gamma\left(\sum_{r=1}^{V}\beta_r\right)}{\prod_{r=1}^{V}\Gamma(\beta_r)}\right)^{K}\prod_{i=1}^{K}\prod_{r\neq v}\Gamma\left(n^i_{(\cdot),r}+\beta_r\right)\frac{\prod_{i=1}^{K}\Gamma\left(n^i_{m,(\cdot)}+\alpha_i\right)}{\Gamma\left(\sum_{i=1}^{K}n^i_{m,(\cdot)}+\alpha_i\right)}\prod_{i=1}^{K}\frac{\Gamma\left(n^i_{(\cdot),v}+\beta_v\right)}{\Gamma\left(\sum_{r=1}^{V}n^i_{(\cdot),r}+\beta_r\right)}\\
&\propto\frac{\prod_{i=1}^{K}\Gamma\left(n^i_{m,(\cdot)}+\alpha_i\right)}{\Gamma\left(\sum_{i=1}^{K}n^i_{m,(\cdot)}+\alpha_i\right)}\prod_{i=1}^{K}\frac{\Gamma\left(n^i_{(\cdot),v}+\beta_v\right)}{\Gamma\left(\sum_{r=1}^{V}n^i_{(\cdot),r}+\beta_r\right)}\\
&\propto\prod_{i=1}^{K}\Gamma\left(n^i_{m,(\cdot)}+\alpha_i\right)\prod_{i=1}^{K}\frac{\Gamma\left(n^i_{(\cdot),v}+\beta_v\right)}{\Gamma\left(\sum_{r=1}^{V}n^i_{(\cdot),r}+\beta_r\right)}.
\end{aligned}
$$

Finally, let $n^{i,-(m,n)}_{j,r}$ be the same meaning as $n^i_{j,r}$ but with the $Z_{(m,n)}$ excluded. The above equation can be further simplified leveraging the property of gamma function. We first split the summation and then merge it back to obtain a k-independent summation, which could be dropped:

$$
\begin{aligned}
&\propto\prod_{i\neq k}\Gamma\left(n^{i,-(m,n)}_{m,(\cdot)}+\alpha_i\right)\prod_{i\neq k}\frac{\Gamma\left(n^{i,-(m,n)}_{(\cdot),v}+\beta_v\right)}{\Gamma\left(\sum_{r=1}^{V}n^{i,-(m,n)}_{(\cdot),r}+\beta_r\right)}\Gamma\left(n^{k,-(m,n)}_{m,(\cdot)}+\alpha_k+1\right)\frac{\Gamma\left(n^{k,-(m,n)}_{(\cdot),v}+\beta_v+1\right)}{\Gamma\left(\sum_{r=1}^{V}n^{k,-(m,n)}_{(\cdot),r}+\beta_r+1\right)}\\
&=\prod_{i\neq k}\Gamma\left(n^{i,-(m,n)}_{m,(\cdot)}+\alpha_i\right)\prod_{i\neq k}\frac{\Gamma\left(n^{i,-(m,n)}_{(\cdot),v}+\beta_v\right)}{\Gamma\left(\sum_{r=1}^{V}n^{i,-(m,n)}_{(\cdot),r}+\beta_r\right)}\Gamma\left(n^{k,-(m,n)}_{m,(\cdot)}+\alpha_k\right)\frac{\Gamma\left(n^{k,-(m,n)}_{(\cdot),v}+\beta_v\right)}{\Gamma\left(\sum_{r=1}^{V}n^{k,-(m,n)}_{(\cdot),r}+\beta_r\right)}\left(n^{k,-(m,n)}_{m,(\cdot)}+\alpha_k+1\right)\frac{n^{k,-(m,n)}_{(\cdot),v}+\beta_v+1}{\sum_{r=1}^{V}n^{k,-(m,n)}_{(\cdot),r}+\beta_r+1}\\
&=\prod_{i}\Gamma\left(n^{i,-(m,n)}_{m,(\cdot)}+\alpha_i\right)\prod_{i}\frac{\Gamma\left(n^{i,-(m,n)}_{(\cdot),v}+\beta_v\right)}{\Gamma\left(\sum_{r=1}^{V}n^{i,-(m,n)}_{(\cdot),r}+\beta_r\right)}\left(n^{k,-(m,n)}_{m,(\cdot)}+\alpha_k+1\right)\frac{n^{k,-(m,n)}_{(\cdot),v}+\beta_v+1}{\sum_{r=1}^{V}n^{k,-(m,n)}_{(\cdot),r}+\beta_r+1}\\
&\propto\left(n^{k,-(m,n)}_{m,(\cdot)}+\alpha_k+1\right)\frac{n^{k,-(m,n)}_{(\cdot),v}+\beta_v+1}{\sum_{r=1}^{V}n^{k,-(m,n)}_{(\cdot),r}+\beta_r+1}
\end{aligned}
$$

Note that the same formula is derived in the article on the Dirichlet-multinomial distribution, as part of a more general discussion of integrating Dirichlet distribution priors out of a Bayesian network.

Related problems [edit]

Related models [edit]

Topic modeling is a classic solution to the problem of information retrieval using linked data and semantic web technology [10]. Related models and techniques are, among others, latent semantic indexing, independent component analysis, probabilistic latent semantic indexing, non-negative matrix factorization, and Gamma-Poisson distribution.

The LDA model is highly modular and can therefore be easily extended. The main field of interest is modeling relations between topics. This is achieved by using another distribution on the simplex instead of the Dirichlet. The Correlated Topic Model[11] follows this approach, inducing a correlation structure between topics by using the logistic normal distribution instead of the Dirichlet. Another extension is the hierarchical LDA (hLDA),[12] where topics are joined together in a hierarchy by using the nested Chinese restaurant process, whose structure is learnt from data. LDA can also be extended to a corpus in which a document includes two types of information (e.g., words and names), as in the LDA-dual model.[13] Nonparametric extensions of LDA include the hierarchical Dirichlet process mixture model, which allows the number of topics to be unbounded and learnt from data.

As noted earlier, pLSA is similar to LDA. The LDA model is essentially the Bayesian version of pLSA model. The Bayesian formulation tends to perform better on small datasets because Bayesian methods can avoid overfitting the data. For very large datasets, the results of the two models tend to converge. One difference is that pLSA uses a variable d to represent a document in the training set. So in pLSA, when presented with a document the model hasn't seen before, we fix $\Pr(w \mid z)$—the probability of words under topics—to be that learned from the training set and use the same EM algorithm to infer $\Pr(z \mid d)$—the topic distribution under d. Blei argues that this step is cheating because you are essentially refitting the model to the new data.

Spatial models [edit]

In evolutionary biology, it is often natural to assume that the geographic locations of the individuals observed bring some information about their ancestry. This is the rational of various models for geo-referenced genetic data[7] [14]

Variations on LDA have been used to automatically put natural images into categories, such as "bedroom" or "forest", by treating an image as a document, and small patches of the image as words;[15] one of the variations is called Spatial Latent Dirichlet Allocation.[16]

See also [edit]

- Variational Bayesian methods
- Pachinko allocation
- tf-idf

References [edit]

1. ^ a b Pritchard, J. K.; Stephens, M.; Donnelly, P. (June 2000). "Inference of population structure using multilocus genotype data". *Genetics*. **155** (2): *pp.* 945–959. ISSN 0016-6731. PMC 1461096. PMID 10835412.
2. ^ Falush, D.; Stephens, M.; Pritchard, J. K. (2003). "Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies". *Genetics*. **164** (4): *pp.* 1567–1587.
3. ^ a b c Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John (ed.). "Latent Dirichlet Allocation". *Journal of Machine Learning Research*. **3** (4–5): *pp.* 993–1022. doi:10.1162/jmlr.2003.3.4-5.993. Archived from the original on 2012-05-01. Retrieved 2006-12-19.
4. ^ Girolami, Mark; Kaban, A. (2003). *On an Equivalence between PLSI and LDA*. Proceedings of SIGIR 2003. New York:

6. ^ Alexander, David H.; Novembre, John; Lange, Kenneth (2009). "Fast model-based estimation of ancestry in unrelated individuals". *Genome Research*. **19** (9): 1655–1664. doi:10.1101/gr.094052.109. PMC 2752134. PMID 19648217.
7. ^ a b Guillot, G.; Estoup, A.; Mortier, F.; Cosson, J. (2005). "A spatial statistical model for landscape genetics". *Genetics*. **170** (3): *pp.* 1261–1280. doi:10.1534/genetics.104.033803. PMC 1451194. PMID 15520263.
8. ^ Minka, Thomas; Lafferty, John (2002). *Expectation-propagation for the generative aspect model* (PDF). Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann. ISBN 1-55860-897-4.
9. ^ Yao, Limin; Mimno, David; McCallum, Andrew (2009). *Efficient methods for topic model inference on streaming document*

11. ^ Blei, David M.; Lafferty, John D. (2006). "Correlated topic models" (PDF). *Advances in Neural Information Processing Systems*. **18**.
12. ^ Blei, David M.; Jordan, Michael I.; Griffiths, Thomas L.; Tenenbaum, Joshua B (2004). *Hierarchical Topic Models and the Nested Chinese Restaurant Process* (PDF). Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference. MIT Press. ISBN 0-262-20152-6.
13. ^ Shu, Liangcai; Long, Bo; Meng, Weiyi (2009). *A Latent Topic Model for Complete Entity Resolution* (PDF). 25th IEEE International Conference on Data Engineering (ICDE 2009).
14. ^ Guillot, G.; Leblois, R.; Coulon, A.; Frantz, A. (2009). "Statistical methods in spatial genetics". *Molecular Ecology*. **18** (23): *pp.* 4734–4756. doi:10.1111/j.1365-294X.2009.04410.x. PMID 19878454.

Association for Computing Machinery. ISBN 1-58113-646-3.

5. ^ Griffiths, Thomas L.; Steyvers, Mark (April 6, 2004). "Finding scientific topics". *Proceedings of the National Academy of Sciences*. **101** (Suppl. 1): 5228–5235. Bibcode:2004PNAS..101.5228G. doi:10.1073/pnas.0307752101. PMC 387300. PMID 14872004.

collections. 15th ACM SIGKDD international conference on Knowledge discovery and data mining.

10. ^ Lamba, Manika; Madhusudhan, Margam (2019). "Mapping of topics in DESIDOC Journal of Library and Information Technology, India: a study". **120**: 477–505. doi:10.1007/s11192-019-03137-5.

15. ^ Li, Fei-Fei; Perona, Pietro. "A Bayesian Hierarchical Model for Learning Natural Scene Categories". *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. **2**: 524–531.

16. ^ Wang, Xiaogang; Grimson, Eric (2007). "Spatial Latent Dirichlet Allocation" (PDF). *Proceedings of Neural Information Processing Systems Conference (NIPS)*.

External links [edit]

This article's **use of external links may not follow Wikipedia's policies or guidelines**. Please improve this article by removing excessive or inappropriate external links, and converting useful links where appropriate into footnote references. *(June 2016) (Learn how and when to remove this template message)*

- jLDADMM A Java package for topic modeling on normal or short texts. jLDADMM includes implementations of the LDA topic model and the *one-topic-per-document* Dirichlet Multinomial Mixture model. jLDADMM also provides an implementation for document clustering evaluation to compare topic models.
- STTM A Java package for short text topic modeling (https://github.com/qiang2100/STTM). STTM includes these following algorithms: Dirichlet Multinomial Mixture (DMM) in conference KDD2014, Biterm Topic Model (BTM) in journal TKDE2016, Word Network Topic Model (WNTM) in journal KAIS2018, Pseudo-Document-Based Topic Model (PTM) in conference KDD2016, Self-Aggregation-Based Topic Model (SATM) in conference IJCAI2015, (ETM) in conference PAKDD2017, Generalized P´olya Urn (GPU) based Dirichlet Multinomial Mixturemodel (GPU-DMM) in conference SIGIR2016, Generalized P´olya Urn (GPU) based Poisson-based Dirichlet Multinomial Mixturemodel (GPU-PDMM) in journal TIS2017 and Latent Feature Model with DMM (LF-DMM) in journal TACL2015. STTM also includes six short text corpus for evaluation. STTM presents three aspects about how to evaluate the performance of the algorithms (i.e., topic coherence, clustering, and classification).
- Lecture that covers some of the notation in this article: LDA and Topic Modelling Video Lecture by David Blei or same lecture on YouTube
- D. Mimno's LDA Bibliography An exhaustive list of LDA-related resources (incl. papers and some implementations)
- Gensim, a Python+NumPy implementation of online LDA for inputs larger than the available RAM.
- topicmodels and lda are two R packages for LDA analysis.
- "Text Mining with R" including LDA methods, video presentation to the October 2011 meeting of the Los Angeles R users group
- MALLET Open source Java-based package from the University of Massachusetts-Amherst for topic modeling with LDA, also has an independently developed GUI, the Topic Modeling Tool
- LDA in Mahout implementation of LDA using MapReduce on the Hadoop platform
- Latent Dirichlet Allocation (LDA) Tutorial for the Infer.NET Machine Computing Framework Microsoft Research C# Machine Learning Framework
- LDA in Spark: Since version 1.3.0, Apache Spark also features an implementation of LDA
- LDA, exampleLDA MATLAB implementation

v · t · e **Natural language processing** [show]

Categories: Statistical natural language processing | Latent variable models | Probabilistic models

This page was last edited on 28 April 2020, at 02:06 (UTC).