Latent Dirichlet Allocation
David M. Blei, Andrew Y. Ng and Michael I. Jordan

Variational Extensions to EM and Multinomial PCA
Wray Buntine

Summary by Jaakko Peltonen, Feb 17, 2004

1 Introduction

The two papers summarized here both consider the task of clustering or modeling discrete data like text documents. In brief, Latent Diriclet Allocation (LDA), introduced by Blei et al [1], is a generative model where the data is generated from combinations of latent distributions or topics. Buntine [2] later gave a more general interpretation of the model under the name Multinomial PCA.

2 Latent Dirichlet Allocation

In Latent Dirichlet Allocation, each document is generated by a two-step process: **1)** Sample a K-dimensional vector θ of multinomial probabilities from a Dirichlet distribution with parameters α. **2)** for each word in the document, first sample a topic z_n with probabilities Multinomial(θ), that is, $p(z_n = k) = \theta_k$. Then sample the actual word w_n with probabilities $p(w_n|z_n)$, which are parameterized as a $k \times |V|$ matrix β.

An alternative view is that the first step samples a particular weighted average (convex combination) of the word probabilities in the topics, and the words are then generated from that distribution.

Notice that the document is not generated from a single topic: the topic is sampled anew for each word—however, the topic proportions are sampled once per document. This yields the following likelihood for a document with word vector $\mathbf{w}$:

$$p(\mathbf{w}) = \int_\theta \left(\prod_{n=1}^{N} \sum_{z_n=1}^{k} p(w_n|z_n;\beta)p(z_n|\theta) \right) p(\theta;\alpha)d\theta \ . \tag{1}$$

Related models. LDA is related to a simple *mixture of unigrams model*, where each document is generated from a single topic. Such a process yields the following likelihood:

$$p(\mathbf{w}) = \sum_{z=1}^{k} \left(\prod_{n=1}^{N} p(w_n|z) \right) p(z) \ . \tag{2}$$

However, this simpler model only has one parameter less than LDA.

Another related model is *probabilistic Latent Semantic Indexing* (pLSI). There, the document index and its words are independent given the topic:

$$p(d, w) = \sum_{z=1}^{k} p(w|z)p(z|d)p(d) \tag{3}$$

where $p(z|d)$ is the topic distribution in the document. This model allows multiple topics per document; however, since d is just a document index, the learning may overfit the training documents without tempering heuristics.

Variational inference. The likelihood (1) is too complex to compute or optimize directly. Instead, Blei et al. use variational approximation, i.e. they optimize a lower bound of the likelihood:

$$\begin{aligned}\log p(\mathbf{w}; \alpha, \beta) &= \log \int_\theta \sum_{\mathbf{z}} p(\mathbf{w}|\mathbf{z}; \beta)p(\mathbf{z}|\theta)p(\theta; \alpha)\frac{q(\theta, \mathbf{z}; \gamma, \phi)}{q(\theta, \mathbf{z}; \gamma, \phi)}d\theta \\ &\geq E_q\{\log p(\mathbf{w}|\mathbf{z}; \beta) + \log p(\mathbf{z}|\theta) + \log p(\theta; \alpha) - \log q(\theta, \mathbf{z}; \gamma, \phi)\} \end{aligned} \tag{4}$$

where q is an approximation for the distribution of the hidden (latent) data. Here $q(\theta, \mathbf{z}; \gamma, \phi) = q(\theta; \gamma)\prod_n q(z_n; \phi_n)$ is a factorized distribution where θ is Dirichlet-distributed with parameters γ and $\mathbf{z}$ are Multinomial-distributed with parameters ϕ. For the whole data $\mathcal{D}$ the log-likelihood is bounded as

$$\log p(\mathcal{D}) \geq \sum_{m=1}^{M} \left(E_{q_m}\{\log p(\theta, \mathbf{z}, \mathbf{w})\} - E_{q_m}\{\log q_m(\theta, \mathbf{z})\}\right) . \tag{5}$$

This bound is optimized by an Expectation Maximization (EM) algorithm where the E-step is variational. That is, in the E step the approximation is optimized by alternating the equations

$$\begin{aligned}\phi_{ni} &\propto \beta_{iw_n} \exp(\Psi_(\gamma_i) - \Psi(\textstyle\sum_{j=1}^{k} \gamma_j)) \\ \gamma_i &= \alpha_i + \textstyle\sum_{n=1}^{N} \phi_{ni}\end{aligned} \tag{6}$$

and in the M step we set $\beta_{ij} \propto \sum_{m=1}^{M}\sum_{n=1}^{|\mathbf{w}_m|} \phi_{mni}w_{mn}^j$ and optimize the α_i by the Newton-Raphson method.

Experiments. Blei et al. test LDA in three tasks: language modeling, document classification and collaborative filtering. In the first task, the quality measure is *perplexity*, which is is inverse to the per-word likelihood, and is defined as $\text{perplexity}(\mathcal{D}_{\text{test}}) = \exp(-\sum_m \log p(\mathbf{w}_m)/\sum_m |\mathbf{w}_m|)$. Results on the AP and CRAN corpora are shown in Fig. 1. LDA outperformed pLSI and mixture of unigrams models. Given the model, Blei et al. were also able to study the topics in a particular document (simply find the topics with largest γ_i, and the corresponding word distributions from β).

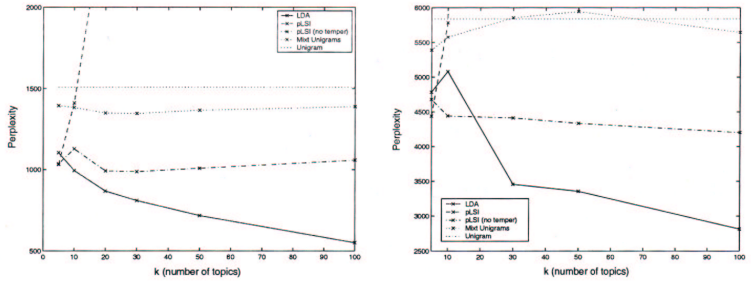

Figure 1: Perplexity results on the AP (left) and CRAN (right) corpora.

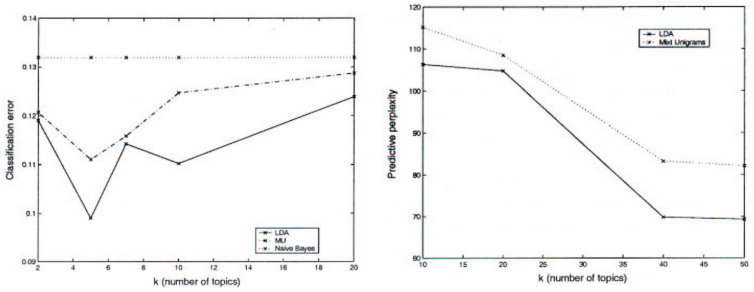

Figure 2: Left: classification results on WebKB data. Right: collaborative filtering results on EachMovie data.

In the second task, a separate model $p(\mathbf{w}|c)$ was learnt for each class, and for new documents the classification was chosen by Bayes' rule ($\arg\max_c p(\mathbf{w}|c)p(c)$). Results on the WebKB dataset are shown in Fig. 2 (left). LDA outperformed mixture of unigrams and Naive Bayes.

In the third task, users from the EachMovie dataset have indicated preferred movies; the task is to predict, for each new user, one unknown preference based on their other preferences. Results are shown in Fig. 2 (right). LDA outperformed mixture of unigrams.

3 A New Interpretation: Multinomial PCA

Buntine [2] gives a different interpretation to LDA and related models from the viewpoint of modeling variation. He starts by making an analogue to Principal Component Analysis (PCA), which can be interpreted as a generative model for continuous data $\mathbf{x}$.

PCA can be seen as the solution to optimizing the following model for $\mathbf{x}$:

$$\begin{aligned} \mathbf{m} &\sim \text{Gaussian}(\mathbf{0}, \mathbf{I}_K) \\ \mathbf{x} &\sim \text{Gaussian}(\mathbf{\Omega m} + \mu, \mathbf{I}_J) \end{aligned} \tag{7}$$

This is a two-step generative process where the hidden (latent) variable $\mathbf{m}$ is sampled first and the observed features $\mathbf{x}$ are sampled with probabilities that depend on $\mathbf{m}$. Here $\mathbf{m}$ effectively creates Gaussian variation along the K directions specified by the matrix $\mathbf{\Omega}$, and $\mathbf{x}$ adds extra Gaussian variation in all J directions. When the parameters of the model are optimized for a particular set of observations, $\mathbf{\Omega}$ will contain the 'principal components' of the distribution of $\mathbf{x}$. In practice they can also be solved as an eigenvalue problem.

For discrete data (also called $\mathbf{x}$), there is a similar generative process:

$$\begin{array}{l} \mathbf{m} \sim \text{Dirichlet}(\alpha) \quad \text{or} \quad \mathbf{m} \sim \text{Entropic}(\lambda) \\ \mathbf{x} \sim \text{Multinomial}(\mathbf{\Omega m}, L) \end{array} \tag{8}$$

This is also a two-step process; the hidden (latent) variable $\mathbf{m}$ creates Dirichlet or Entropic variation in the region between the prototype distributions contained in $\mathbf{\Omega}$, and $\mathbf{x}$ adds extra variation because of the Multinomial sampling for L discrete words.

Deriving clustering algorithms. Buntine next develops clustering algorithms that optimize the above discrete-data model. The optimization is based on a *variational extension* of the standard Expectation Maximization (EM) algorithm. The idea is that although the model is too complex to compute the marginal likelihood of the data and the parameters directly, one can optimize a *lower bound* for it.

The bound is based on a *Kullback-Leibler approximation* (also called mean-field approximation) where a simpler distribution for the hidden variables, denoted $q(\mathbf{h}_{\{\}}|\theta)$, is used to approximate their true distribution p. The bound is

$$\begin{aligned} \log p(\mathbf{x}_{\{\}}, \phi) - KL(q(\mathbf{h}_{\{\}}|\theta)||p(\mathbf{h}_{\{\}}|\mathbf{x}_{\{\}}, \phi)) \\ = E_q(\mathbf{h}_{\{\}}|\theta)\{\log p(\mathbf{x}_{\{\}}, \mathbf{h}_{\{\}}, \phi)\} + H(q(\mathbf{h}_{\{\}}|\theta) \ , \end{aligned} \tag{9}$$

where ϕ are the parameters of the true distribution and θ are the parameters of the approximation. Notice that on the first line, only the second term depends on θ, and on the second line, only the first term depends on ϕ.

The optimization is done by alternating two steps: **1)** make the approximation q as close as possible to the true distribution (in the sense of minimal Kullback-Leibler divergence), and **2)** optimize the parameters of p based on the approximation. If the approximation is flexible enough to reach the true distribution, this kind of optimization is simply the standard EM algorithm.

Optimizing the approximation. The precise form of the first step depends on the approximation q: if it is from the exponential family, the parameter update is

$$\theta \leftarrow \frac{\partial}{\partial \mu_t} E_{q(\mathbf{h}_{\{\}}|\theta)}\{\log p(\mathbf{h}_{\{\}}|\mathbf{x}_{\{\}}, \phi) + \log Y_t(\mathbf{h}_{\{\}})\} \tag{10}$$

where Y_t is part of q and μ_t are *dual parameters* for θ. On the other hand, if the approximation is a factorized distribution $q(\mathbf{h}_{\{\}}) = q_1(\mathbf{h}_{\{\},1})q_2(\mathbf{h}_{\{\},2})$the update is

$$\begin{aligned} q_1(\mathbf{h}_{\{\},1}) &\leftarrow \exp(E_{q_2(\mathbf{h}_{\{\},2})}\{\log p(\mathbf{x}|\phi)\}) \\ q_2(\mathbf{h}_{\{\},2}) &\leftarrow \exp(E_{q_1(\mathbf{h}_{\{\},1})}\{\log p(\mathbf{x}|\phi)\}) \end{aligned} \tag{11}$$

If the approximation is factorized and from the exponential family, both forms are equal.

The final algorithms. Buntine considers several priors for the hidden variable $\mathbf{m}$ (using the notation from the beginning of this Section). Two Dirichlet priors, a hierarchical prior, and an entropic prior are all possible.

There are also two different cases for the data likelihood: whether ordering of words is relevant or not. They are identical except for a combinatorial term which cancels out in the optimization of the approximation. The model for generating a document of length L is

$$\begin{aligned} &\mathbf{m} \sim \text{Dirichlet}(\alpha) \\ &\mathbf{c} \sim \text{Multinomial}(\mathbf{m}, L) \\ &\mathbf{w}_{k,\cdot} \sim \text{Multinomial}(\mathbf{\Omega}_{k,\cdot}, c_k) \quad \text{for } k = 1, \ldots, K \end{aligned} \tag{12}$$

where $\mathbf{w}$ is a matrix of hidden components versus words, whose element (k, j) tells how many words j were generated from component k. The actual observation vector $\mathbf{r}$ is the sum of $\mathbf{w}$ over rows.

Buntine presents two algorithms for optimizing this model. The first optimizes a lower bound of $\log p(\mathbf{\Omega}, \alpha|\mathbf{r})$, when a factorized approximation $q(\mathbf{m})q(\mathbf{w})$ is used for the hidden variable distribution $p(\mathbf{m}, \mathbf{w}|\mathbf{\Omega}, \alpha, \mathbf{r})$. The algorithm is

$$\begin{aligned} &\gamma_{j,k,[i]} \leftarrow \tfrac{1}{Z_{3,j,[i]}}\Omega_{k,j}\exp(\Psi_0(\beta_{k,[i]}) - \Psi_0(\textstyle\sum_k \beta_{k,[i]})) \\ &\beta_{k,[i]} \leftarrow \alpha_k + \textstyle\sum_j r_{j,[i]}\gamma_{j,k,[i]} \\ &\Omega_{k,j} \leftarrow \tfrac{1}{Z_{4,k}}(2f_j + \textstyle\sum_i r_{j,[i]}\gamma_{j,k,[i]}) \\ &\Psi_0(\alpha_k) - \Psi_0(\textstyle\sum_k \alpha_k) \leftarrow \tfrac{1}{1+I}\left(\log\tfrac{1}{K} + \textstyle\sum_i \Psi_0(\beta_{k,[i]}) - \Psi_0(\textstyle\sum_k \beta_{k,[i]})\right) \end{aligned} \tag{13}$$

where γ and β are parameters of the approximation q, the Z are various normalization terms from the distributions, and the last line uses the dual form of α. This algorithm is an extension of Blei et al.:s LDA, with a prior for $\mathbf{\Omega}$ and simpler handling of the Dirichlet parameters.

The second algorithm (not shown here) instead optimizes a lower bound of $\log p(\mathbf{\Omega}, \alpha, \mathbf{m}|\mathbf{r})$, i.e., a single value of the hidden variable $\mathbf{m}$ is used. It is equivalent to pLSI and the Nonnegative Matrix Factorization (NMF; [3]) algorithm.

Experiments. Buntine does not make additional comparisons but instead studies the properties of the algorithm on two kinds of data: bag-or-words document data (the Reuters-21578 dataset) and bigram data (Google Bigrams). Three properties are studied: expected components (EC), expected words per

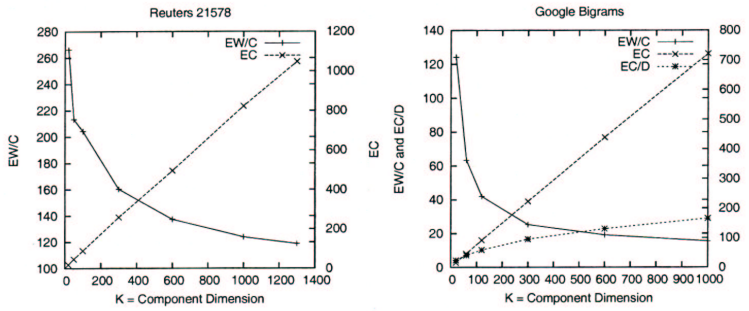

Figure 3: Properties of components in the Multinomial PCA algorithm.

component (EW/D), and expected components per document (EC/D); the results are shown in Fig. 3.

Buntine finds that on the document data, a newswire (document) typically belongs to 2 components, but on the Google Bigram data one word belongs to several components depending on sample size. He also finds that the use of priors for the parameters α led to a better match between the expected and observed component proportions than the maximum likelihood estimates of Blei et al. Lastly, he finds that the components in the bigram data *unfold* as more components are added: for example, general forms like verbs break into people verbs.

References

[1] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[2] W. Buntine. Variational Extensions to EM and Multinomial PCA. In T. Elomaa, H. Mannila, H. Toivonen, editors, *Machine Learning: ECML 2002*, pages 23-34, Springer-Verlag Heidelberg, 2002.

[3] Lee, D., Seung, H. Learning the parts of objects by non-negative matrix factorization. *Nature* **401**, pages 50-57, 1999.