

# Variational Extensions to EM and Multinomial PCA

Wray Buntine

Helsinki Institute of Information Technology  
P.O. Box 9800, FIN-02015 HUT, Finland  
[wray.buntine@hiit.fi](mailto:wray.buntine@hiit.fi)  
<http://www.hiit.fi/wray.buntine>

**Abstract.** Several authors in recent years have proposed discrete analogues to principle component analysis intended to handle discrete or positive only data, for instance suited to analyzing sets of documents. Methods include non-negative matrix factorization, probabilistic latent semantic analysis, and latent Dirichlet allocation. This paper begins with a review of the basic theory of the variational extension to the expectation-maximization algorithm, and then presents discrete component finding algorithms in that light. Experiments are conducted on both bigram word data and document bag-of-words to expose some of the subtleties of this new class of algorithms.

## 1 Introduction

In text and image analysis, standard clustering algorithms are unsatisfactory because documents or images seem to mix additively in contrast to the mutually exclusive mixture semantics of standard clustering. Principle components analysis (PCA) is unsatisfactory because it has a multivariate Gaussian interpretation [1] difficult to justify with the low discrete counts for words in documents, and it comes up with difficult to interpret components, e.g., with negative quantities. Its cousin, latent semantic indexing (LSI), also has interpretation problems due to its Gaussian nature [2]. Authors have proposed analogues to PCA intended to handle discrete or positive only data. Methods include non-negative matrix factorization (NMF) [3], probabilistic latent semantic analysis (pLSI) [2] latent Dirichlet allocation (LDA) [4], and a general purpose extension of PCA itself to Bregman distances [5], which are a generalization of Kullback-Leibler (KL) divergence. A good discussion of the motivation for these techniques can be found in [2], and an analysis of related reduced dimension models and some of the earlier statistical literature which used simpler algorithms can be found in [6]. Related models using Dirichlets have been dubbed Dirichlet mixtures and applied extensively in molecular biology [7].

A common problem with the earlier formulations of these discrete component analysis [3, 2, 5] is that they fail to make a full probability model of the target data in question, a model where hidden variables, observed data, and assumptions are all clearly exposed. Moreover, the relationship to LDA remains

unclear. In this paper I present the problem as a multinomial analogue to PCA. However, unlike standard PCA, spectral analysis does not come to the rescue to yield a simple solution. Instead, the usual statistical machinery for mixture distributions needs to be wheeled out, and applied. This paper reviews the EM algorithm and its variational extension, and applies them to multinomial PCA.

## 2 The Problem of Document Components

Consider Tipping *et al.*'s [1] representation of standard PCA. The resultant algorithm is a simple application of numerical methods: find the largest  $k$  eigenvectors. Their span represents a projection of the data containing most of the “variance.”

A hidden variable  $\mathbf{m}$  is sampled from  $K$ -dimensional Gaussian noise. Each entry represents the strength of the corresponding component and can be positive or negative. This is folded with the  $J \times K$  matrix of component means  $\mathbf{\Omega}$  and then used as a mean of  $J$ -dimensional Gaussian. For documents represented as a bag of words,  $J$  would represent the number of words in the application's dictionary and is expected to be considerably larger than  $K$ .

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

Satellite and telescope images, for instance, are often analyzed as “pure” high resolution, positive pixel elements added to form a lower resolution pixel, thus a convex combination of components is a suitable model of pixel types. Likewise, the Poisson distribution is more appropriate for the small counts seen in some telescope image data. Note that if the total sum for a set of independent Poisson variables is known, then their joint distribution becomes multinomial. The multinomial is used here as the basic discrete distribution.

A discrete analogue to the above Gaussian formulation is first to sample a probability vector  $\mathbf{m}$  that represents the proportional weighting of components, and then to mix it with a set of probability vectors  $\mathbf{\Omega}$  representing the component means:

$$\begin{aligned}\mathbf{m} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \quad \text{or} \quad \mathbf{m} \sim \text{Entropic}(\lambda) , \\ \mathbf{x} &\sim \text{Multinomial}(\mathbf{\Omega}\mathbf{m}, L) ,\end{aligned}$$

where  $L$  is the total number of words in the document. By varying the distribution on  $\mathbf{m}$  the model can have different behaviors. The Dirichlet has a vector of  $K$ -dimensional parameters,  $\boldsymbol{\alpha}$  and the entropic prior used here is an extension of Brand's [8] where  $p(\mathbf{m}) \propto \exp(-\lambda H(\mathbf{m}))$ . This analogue is an example of Collins *et al.*'s [5] generalization of PCA.

For the Gaussian case,  $\mathbf{\Omega}$  can be folded into the covariance matrix leaving a mixture problem directly amenable to solution via the EM algorithm [9], but also simplified into a “ $K$ -th largest eigenvectors” problem. However, for the multinomial case, there appears to be no such transformation. So more sophisticated mixture modeling is needed.

### 3 Background Theory

The theory of exponential family distributions and Kullback-Leibler approximations is briefly reviewed here. The formulations of Ghahramani and Beal [10] and Buntine [11] are roughly followed. A notation convention used here is that indices  $i, j, k, l$  in sums and products always range over 1 to  $I, J, K, L$  respectively.  $i$  usually denotes a sample index,  $j$  a dictionary word index,  $k$  a component index, and  $l$  a word-in-document index.

#### 3.1 Exponential Family Distributions

The general exponential family distribution takes the form as follows. For an individual sample point, a vector of measurements  $\mathbf{x}$ , we have a vector of  $T$  functions  $\mathbf{t}(\mathbf{x})$  and some parameters  $\boldsymbol{\theta}$  also of dimension  $T$ , and possibly with some additional constraints. The probability  $q(\mathbf{x} | \boldsymbol{\theta})$  has the form

$$q(\mathbf{x} | \boldsymbol{\theta}) = \frac{1}{Y_t(\mathbf{x})Z_t(\boldsymbol{\theta})} \exp(\mathbf{t}(\mathbf{x})^\dagger \boldsymbol{\theta}) .$$

I also usually abbreviate  $Z_t(\boldsymbol{\theta})$  to  $Z$  or add a distinguishing subscript.

I use the notation  $E_{q(y|\phi)}\{A\}$  to denote the expected value of the quantity  $A$  when  $y$  is distributed according to  $q(y|\phi)$ . Two key definitions needed [11] are:

$$\begin{aligned} \boldsymbol{\mu}_t &\equiv E_{q(\mathbf{x}|\boldsymbol{\theta})}\{\mathbf{t}(\mathbf{x})\} = \frac{\partial \log Z_t}{\partial \boldsymbol{\theta}} , \\ \boldsymbol{\Sigma}_t &\equiv E_{q(\mathbf{x}|\boldsymbol{\theta})}\{(\mathbf{t}(\mathbf{x}) - \boldsymbol{\mu}_t)(\mathbf{t}(\mathbf{x}) - \boldsymbol{\mu}_t)^\dagger\} = \frac{\partial^2 \log Z_t}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{\mu}_t}{\partial \boldsymbol{\theta}} . \end{aligned} \quad (1)$$

The mean vector  $\boldsymbol{\mu}_t$  has the same dimensionality as  $\boldsymbol{\theta}$  and the matrix  $\boldsymbol{\Sigma}_t$  is the covariance of  $\mathbf{t}(\mathbf{x})$ . If  $\boldsymbol{\theta}$  is in a region where  $\boldsymbol{\Sigma}_t$  is not of full rank, then the data vector  $\mathbf{t}(\mathbf{x})$  is redundant. Thus the following remarkable condition holds:  $\boldsymbol{\mu}_t$  can be treated as a dual set of parameters to  $\boldsymbol{\theta}$ . When  $\boldsymbol{\Sigma}_t$  is of full rank, it is the Hessian for the change of basis. While  $\boldsymbol{\Sigma}_t$  is also the expected Fisher Information for the distribution. Note that both  $\boldsymbol{\mu}_t$  and  $\boldsymbol{\Sigma}_t$  are directly derivable from  $Z_t$ .

For a univariate Gaussian with mean  $\mu$  and standard deviation  $\sigma$ ,  $\mathbf{t}(\mathbf{x}) = (x, x^2)$ ,  $\boldsymbol{\theta} = (\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2})$  and  $\boldsymbol{\mu}_t = (\mu, \sigma^2 + \mu^2)$ . Examples of this form for the multinomial distribution with probability vector  $\boldsymbol{\alpha}$  and count  $N$  and the Dirichlet distribution with probability vector  $\boldsymbol{\alpha}$  are given in Table 1. Here  $\Gamma(y)$  is the gamma function,  $\Psi_0(y)$  is the digamma function, and the parameter vector for the multinomial has the constraint that  $\sum_k \alpha_k = 1$ . Note that for the Dirichlet, the Hessian  $\boldsymbol{\Sigma}_t$  is of full rank when each  $\alpha_k > 0$  and computing the parameter vector  $\boldsymbol{\alpha}$  from the dual can be done using Minka's fixed point method [12].

One more key result about the exponential family is computing maximum *a posteriori* (MAP) values for parameters given a sample of  $I$  data points. From the likelihood, all that matters is the so-called sufficient statistics:  $\sum_i \mathbf{t}(\mathbf{x}_i)$ . A conjugate prior has the same functional form as the likelihood. One way to model these is to have an "effective" prior sample whose sufficient statistics

**Table 1.** Exponential family characterizations

$Z_t$	$t_k(\mathbf{x})$	$\theta_k$	$\mu_{t,k}$
1	$x_k$	$\log \alpha_k$	$N\alpha_k$
$\frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$	$\log r_k$	$\alpha_k - 1$	$\Psi_0(\alpha_k) - \Psi_0(\sum_k \alpha_k)$

are  $\nu_t$  and the prior sample size is  $S_t$ . In this case, the unique MAP for the exponential family parameters yields an estimate for the dual

$$\widehat{\boldsymbol{\mu}}_t = \frac{\boldsymbol{\nu}_t + \sum_i \mathbf{t}(\mathbf{x}_i)}{S_t + I} . \quad (2)$$

### 3.2 Kullback-Leibler Approximations

Consider a distribution  $p(\mathbf{x} | \phi)$  which is a posterior rendered impractical for use due to normalization or marginalization problems. Approximate it with a distribution of the form  $q(\mathbf{x} | \theta)$ . The so-called ‘‘mean-field’’ approximation is to choose  $\theta$  by minimizing the KL divergence between distributions  $q$  and  $p$ ,

$$KL(q(\mathbf{x} | \theta) || p(\mathbf{x} | \phi)) = E_{q(\mathbf{x} | \theta)} \left\{ \log \frac{q(\mathbf{x} | \theta)}{p(\mathbf{x} | \phi)} \right\} ,$$

On the RHS,  $p(\mathbf{x}, \phi)$  can replace  $p(\mathbf{x} | \phi)$ , and their normalizing constants can be ignored.

**Kullback-Leibler Approximations on Exponential Family.** Suppose  $q(\mathbf{x} | \theta)$  is in the exponential family as described in Section 3. For this minimization task, the following fixed-point update formula can be used to optimize for  $\theta$ :

$$\boldsymbol{\theta} \leftarrow \frac{\partial}{\partial \boldsymbol{\mu}_t} (E_{q(\mathbf{x} | \theta)} \{ \log p(\mathbf{x} | \phi) + \log Y_t(\mathbf{x}) \}) . \quad (3)$$

*Proof. (sketch)* Substitute the expected  $\log q()$  term in the KL with the entropy  $H(q(\mathbf{x} | \theta))$  given by  $E_{q(\mathbf{x} | \theta)} \{ \log Y_t(\mathbf{x}) \} + \log Z_t - \boldsymbol{\mu}_t^\dagger \boldsymbol{\theta}$  and differentiate w.r.t.  $\boldsymbol{\mu}_t$ .  $\square$

Note that by Eqn. (1),  $\boldsymbol{\mu}_t$  often occurs linearly in the expected value thus this can be easy. For the Gaussian and multinomial, the dual parameters  $\boldsymbol{\mu}_t$  are the ones we are familiar with anyway.

**Kullback-Leibler Approximations on Products.** Another view of these approximations can be gained by looking at a factored distribution [10]<sup>1</sup>. Consider approximating  $p(\mathbf{x} | \phi)$  by a distribution that factors  $\mathbf{x}$  into two independent, non-overlapping components,  $\mathbf{x}_1, \mathbf{x}_2$ :  $q(\mathbf{x}) = q_1(\mathbf{x}_1)q_2(\mathbf{x}_2)$ .

<sup>1</sup> Indeed, they show this also applies to Markov and Bayesian network models

From functional analysis (when  $p()$  is non-zero everywhere), one can show the following minimizes the KL for all independent components of the form above:

$$\begin{aligned} q_1(\mathbf{x}_1) &\leftarrow \frac{1}{Z_1} \exp(E_{q_2(\mathbf{x}_2)} \{\log p(\mathbf{x} | \phi)\}) \\ q_2(\mathbf{x}_2) &\leftarrow \frac{1}{Z_2} \exp(E_{q_1(\mathbf{x}_1)} \{\log p(\mathbf{x} | \phi)\}) \end{aligned} \quad (4)$$

Note this is cyclic,  $q_1()$  is defined in terms of  $q_2()$  and *visa versa*, where the  $\mathbf{x}_1$  and  $\mathbf{x}_2$  terms in the log probability are repeatedly replaced by their mean. This result yields the same rules as (3) when they both apply.

### 3.3 Computational Methods with Hidden Variables

With so-called hidden variables, each data point in the sample is in the form  $\mathbf{x}_{[i]}$  the *observed* data, but there is also unknown  $\mathbf{h}_{[i]}$  the *hidden* data, for  $i = 1, \dots, I$ . The special subscript  $[i]$  is used to denote part of the  $i$ -th data. Denote the full set of these vectors by  $\mathbf{x}_{\{ \}}$  and  $\mathbf{h}_{\{ \}}$ . The observed data is in the exponential family when the hidden data is known, and the hidden data itself is in the exponential family. Suppose the parameter sets for these two distributions are  $\phi_1$  and  $\phi_2$  respectively, yielding  $\phi$  as the full set. I won't flesh out the details of these until the examples later on. The full joint distribution then becomes:

$$p(\phi, \mathbf{x}_{\{ \}}, \mathbf{h}_{\{ \}}) = p(\phi) \prod_i p(\mathbf{h}_{[i]} | \phi_1) p(\mathbf{x}_{[i]} | \mathbf{h}_{[i]}, \phi_2) .$$

Several computational methods present themselves here for maximizing the joint  $p(\phi, \mathbf{x}_{\{ \}})$  where the hidden variables are marginalized out.

**Approximating the MAP Directly with Variational Methods.** It is not known if one can compute the maximum *a posterior* value for  $p(\phi | \mathbf{x}_{\{ \}})$  exactly. Instead consider the following function [13]:

$$\mathcal{L}(\phi; \theta) = \log p(\mathbf{x}_{\{ \}}, \phi) - KL(q(\mathbf{h}_{\{ \}} | \theta) || p(\mathbf{h}_{\{ \}} | \mathbf{x}_{\{ \}}, \phi)) \quad (5)$$

$$= E_{q(\mathbf{h}_{\{ \}} | \theta)} \{ \log p(\mathbf{x}_{\{ \}}, \mathbf{h}_{\{ \}}, \phi) \} + H(q(\mathbf{h}_{\{ \}} | \theta)) . \quad (6)$$

The two lines are easily shown to be equivalent. Note that  $\mathcal{L}()$  represents a lower bound on  $\log p(\mathbf{x}_{\{ \}}, \phi)$ . Thus maximizing it yields a variational algorithm [13]. Repeat the following:

1. Minimize the KL divergence in (5) w.r.t.  $\theta$ .
2. Maximize the expected value in (6) w.r.t.  $\phi$ .

Note the first step would use the methods just developed, Eqns. (3) and (4). The second step is usually done using Eqn. (2) directly because the expected value will look like the log probability of some exponential family likelihoods, thus having a unique global maximum.

**Computing the MAP via EM.** If the exponential family  $q(\mathbf{h}_i | \boldsymbol{\theta}_i, \boldsymbol{\psi})$  is rich enough to include  $p(\mathbf{h}_i | \mathbf{x}_i, \boldsymbol{\phi})$ , then this previous variational method becomes the EM algorithm. In this case,  $KL = 0$  and therefore the MAP for  $\boldsymbol{\phi}$  is obtained. For instance, in standard clustering algorithms,  $\mathbf{h}_i$  is a discrete variable and  $q(\cdot)$  is the discrete distribution on that variable. Thus EM is a variational algorithm at the end condition, where the bound is tight!

## 4 Algorithms for Document Clustering

First I will develop mixture models extending the basic idea of multinomial PCA to be mixture models of convenient exponential family distributions, and then we can grind the formula just presented to produce some algorithms.

### 4.1 Priors for Document Clustering

In the model,  $\mathbf{m}$ , a  $K$ -dimensional probability vector, represents the proportion of components in a particular document.  $\mathbf{m}$  must therefore represent quite a wide range of values with a mean representing the general frequency of components in the sample. Potential priors for a probability vector such as  $\mathbf{m}$  are well discussed in the literature including:

- A Dirichlet prior with equal parameters of 1 or 0.5 (uniform and Jeffreys' prior respectively), or  $C/K$  for prior sample size  $C$ , c.f., Eqn. (2).
- A hierarchical prior, and let the parameters  $\boldsymbol{\alpha}$  for a general Dirichlet be estimated using the data [4].
- The entropic prior [8], which tends to extinguish small components.

For convenience, I will assume the general Dirichlet and specialize to the other forms as required.

For the general Dirichlet prior on the proportions  $\mathbf{m}$ , priors on the  $\boldsymbol{\alpha}$  are needed. We can use a prior for  $\boldsymbol{\alpha}$  corresponding to a prior sample of uniform probability of size 1, and apply Eqn. (2) to compute the MAP for it.

The component means are columns of  $\boldsymbol{\Omega}$  represented as  $\boldsymbol{\Omega}_{k,\cdot}$  for  $k = 1, \dots, K$  are  $J$ -dimensional probability vectors over the  $J$  distinct words, and the prior on them is important to ensure low-count components are handled well. Given a suitable universe of words, one can use an empirical prior for these based on the empirical proportion of words in the universe,  $\mathbf{f}$ , for  $\sum_j f_j = 1$ :  $\boldsymbol{\Omega}_{k,\cdot} \sim \text{Dirichlet}(2\mathbf{f})$ , where 2 represents some small prior sample size.

### 4.2 Likelihoods for Document Clustering

**Where Ordering is Relevant.** In this case, iterate over the  $L$  words as they appear in the document.  $d_l$  is the dictionary index for the  $l$ -th word in the document, where  $d_l$  takes a value from 1 to  $J$ .

$$\begin{aligned} \mathbf{m} &\sim \text{Dirichlet}(\boldsymbol{\alpha}), \\ k_l &\sim \text{Discrete}(\mathbf{m}) && \text{for } l = 1, \dots, L, \\ d_l &\sim \text{Discrete}(\boldsymbol{\Omega}_{k_l,\cdot}) && \text{for } l = 1, \dots, L. \end{aligned}$$

The hidden variables here are  $\mathbf{m}$  and  $\mathbf{k}$  for each document. This turns out to have an identical likelihood to the next case, except for a combinatoric term which, for instance, is canceled out by the  $\log Y_t(x)$  term in Eqn. (3).

**Where Ordering is Irrelevant.** In this case, iterate over word counts which have been totalled for the document. The index  $j = 1, \dots, J$  runs over dictionary index values, and the full hidden data matrix for a single document  $w_{k,j}$  is the count of the number of times the  $j$ -th dictionary word occurs in the document representing the  $k$ -th component. Two derived vectors are the column-wise totals  $c_k$ , the number of words in the document representing the  $k$ -th component, and the row-wise totals  $\mathbf{r}$ , the observed data, typically stored in sparse form. Denote by  $\mathbf{w}_{\cdot,j}$  the  $j$ -th row vector and  $\mathbf{w}_{k,\cdot}$  the  $k$ -th column vector of  $\mathbf{w}$ .

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

The hidden variables here are  $\mathbf{m}$  and  $\mathbf{w}$  for each document. The full likelihood for a single document  $p(\mathbf{m}, \mathbf{w} | \boldsymbol{\alpha}, \boldsymbol{\Omega})$  then simplifies to:

$$\frac{1}{Z_D(\boldsymbol{\alpha})} C_{w_{1,1}, \dots, w_{K,1}, \dots, w_{1,J}, \dots, w_{K,J}}^L \prod_k m_k^{\alpha_k - 1} \prod_{k,j} m_k^{w_{k,j}} \Omega_{k,j}^{w_{k,j}} . \quad (7)$$

Note that  $w_{\cdot,j}$  can be marginalized out (because  $\sum_{k,j} m_k \Omega_{k,j} = 1$ ) yielding the original model proposed in Section 2.

The important aspect of this model is that the hidden variables  $\mathbf{m}$  and  $\mathbf{w}$  remain linked in the likelihood, and thus if  $q(\cdot)$  for the mean field approximation is a product distribution,  $KL$  in Eqn. (5) cannot be zero, so an EM algorithm does not appear feasible.

### 4.3 Multinomial PCA with Dirichlet Prior

The first algorithm here estimates the MAP parameters for  $\boldsymbol{\Omega}$  with the general Dirichlet prior on  $\mathbf{m}$  using parameters  $\boldsymbol{\alpha}$ . This extends [4] with a prior, simpler handling of the Dirichlet parameters, and a proof of optimality of the product approximation.

**Theorem 1.** *Given the likelihood model of Section 4.2 and Eqn. (7), and the priors:  $\mathbf{m} \sim \text{Dirichlet}(\boldsymbol{\alpha})$  and  $\boldsymbol{\Omega}_{k,\cdot} \sim \text{Dirichlet}(2\mathbf{f})$ . The following updates converge to a local maximum of a lower bound of  $\log p(\boldsymbol{\Omega}, \boldsymbol{\alpha} | \mathbf{r})$  that is optimal for all product approximations  $q(\mathbf{m})q(\mathbf{w})$  for  $p(\mathbf{m}, \mathbf{w} | \boldsymbol{\Omega}, \boldsymbol{\alpha}, \mathbf{r})$ . The subscript  $[i]$  indicates values from the  $i$ -th document.*

$$\gamma_{j,k,[i]} \leftarrow \frac{1}{Z_{3,j,[i]}} \Omega_{k,j} \exp \left( \Psi_0(\beta_{k,[i]}) - \Psi_0 \left( \sum_k \beta_{k,[i]} \right) \right) ,$$

$$\beta_{k,[i]} \leftarrow \alpha_k + \sum_j r_{j,[i]} \gamma_{j,k,[i]} , \quad (8)$$

$$\Omega_{k,j} \leftarrow \frac{1}{Z_{4,k}} \left( 2f_j + \sum_i r_{j,[i]} \gamma_{j,k,[i]} \right) ,$$

$$\Psi_0(\alpha_k) - \Psi_0 \left( \sum_k \alpha_k \right) \leftarrow \frac{\log 1/K + \sum_i \Psi_0(\beta_{k,[i]}) - \Psi_0 \left( \sum_k \beta_{k,[i]} \right)}{1 + I} . \quad (9)$$

The exponential in the first equation is an estimate of  $m_k$  as  $\exp(E_q\{\log m_k\})$  which reduces the component entropy  $H(\mathbf{m})$ . Note the last two equations are the standard MAPs for a multinomial and a Dirichlet respectively. The last equation rewrites  $\boldsymbol{\alpha}$  in terms of its dual representation (according to exponential family convention), which is immediately inverted using Minka's methods. The proof is outlined below because it highlights the simplicity of this using the exponential family machinery of Section 3.

*Proof. (sketch)* For this, the first step is a KL approximation to the likelihood of the hidden variables  $p(\mathbf{h}_\Omega | \mathbf{x}_\Omega, \boldsymbol{\phi})$ . The observed data is  $\mathbf{r}$  the row totals of  $\mathbf{w}$ , thus use the product approximation,  $q(\mathbf{m}) \prod_j q(\mathbf{w}_{\cdot,j})$ . Taking an expected value of  $\mathbf{m} \sim q(\mathbf{m})$  over  $\log p(\mathbf{m}, \mathbf{w}, \mathbf{r} | \boldsymbol{\alpha}, \boldsymbol{\Omega})$  yields a form that is an independent multinomial for  $q(\mathbf{w}_{\cdot,j})$  for each  $j$ . For  $\mathbf{w} \sim q(\mathbf{w}_{\cdot,j})$  yields a form that is Dirichlet in  $\mathbf{m}$ . Thus the optimal product distribution has  $\mathbf{m} \sim \text{Dirichlet}(\boldsymbol{\beta})$  and  $\mathbf{w}_{\cdot,j} \sim \text{Multinomial}(\boldsymbol{\gamma}_{j,\cdot}, r_j)$  for some parameter vectors  $\boldsymbol{\beta}$  and  $\boldsymbol{\gamma}$ .

For this problem, either Eqn. (4) or Eqn. (3) works equally well. First, inspect the required log probabilities of Eqn. (4),  $E_{q(\mathbf{m}|\mathbf{r})}\{\log p(\cdot)\}$  and  $E_{q(\mathbf{w}|\mathbf{r})}\{\log p(\cdot)\}$  together with Table 1. Eqn. (4) now becomes, ignoring constants

$$\begin{aligned} \sum_k \log m_k (\beta_k - 1) &\leftarrow \sum_k \log m_k \left( \alpha_k - 1 + \sum_j r_j \gamma_{j,k} \right) , \\ \sum_k w_{k,j} \log \gamma_{j,k} &\leftarrow \sum_k w_{k,j} \left( \Psi_0(\beta_k) - \Psi_0 \left( \sum_k \beta_k \right) + \log \Omega_{k,j} \right) . \end{aligned}$$

And the rewrite rules for  $\boldsymbol{\beta}$  and  $\boldsymbol{\gamma}$  can be extracted using the left hand sides as the updated values.

The second step of the algorithm is to re-estimate the model parameters  $\boldsymbol{\alpha}$  and  $\boldsymbol{\Omega}$  based on optimizing the expected log probability of Eqn. (6). This again can be done by inspection using Table 1 as a guide. Rearrange the full log probability to make it look like posteriors for Dirichlet sampling and multinomial sampling for  $\boldsymbol{\alpha}$  and  $\boldsymbol{\Omega}$  respectively and then apply Eqn. (2) to write down the MAP values. This gives the last two rewrites in the theorem.  $\square$

A simpler version of this theorem is to optimize  $\log p(\mathbf{m}, \boldsymbol{\alpha}, \boldsymbol{\Omega})$  jointly.



**Theorem 2.** *In the context of Theorem 1, the following updates converge to a local maximum of  $\log p(\boldsymbol{\Omega}, \boldsymbol{\alpha}, \mathbf{m} | \mathbf{r})$ .*

$$\begin{aligned} \gamma_{j,k,[i]} &\leftarrow \frac{1}{Z_{5,j,[i]}} \Omega_{k,j} m_{k,[i]} , \\ m_{k,[i]} &\leftarrow \frac{1}{Z_{6,[i]}} \left( \alpha_k - 1 + \sum_j r_{j,[i]} \gamma_{j,k,[i]} \right) , \\ \Omega_{k,j} &\leftarrow \frac{1}{Z_{7,k}} \left( 2f_j + \sum_i r_{j,[i]} \gamma_{j,k,[i]} \right) , \\ \Psi_0(\alpha_k) - \Psi_0 \left( \sum_k \alpha_k \right) &\leftarrow \frac{\log 1/K + \sum_i \log m_{k,[i]}}{1 + I} . \end{aligned}$$

*Proof. (sketch)* Since this case has some hidden variables in the primary objective function, it is not covered by Eqns. (5) and (6). Move  $\mathbf{m}$  across the probability terms to yield a modified formula for variational optimization of the log probability above. For this case  $\boldsymbol{\beta}$  disappears because  $\mathbf{m}$  is fixed as far as the KL approximation is concerned. Optimization for  $\mathbf{m}$  now occurs in the second step of the algorithm. The minimum KL divergence will be zero because  $q(\mathbf{w} | \boldsymbol{\gamma}, \mathbf{r}, \mathbf{m})$  can be exactly modeled with multinomials.  $\square$

#### 4.4 Comparisons

The algorithm of Thm. 2, ignoring priors, is equivalent to the NMF algorithm of Lee and Seung [3] where a final normalization using the  $Z$ 's is not done, and the pLSI algorithm of Hofmann [2], which also includes a regularization step. These correspondences are tricky because, for instance, Hofmann marginalizes  $\boldsymbol{\Omega}$  in the reverse direction. Moreover, the only difference between the algorithms of Thms. 1 and 2 is the estimation of  $m_k$ :  $\exp(E_q\{\log m_k\})$  versus the MAP estimate.

Note if any other prior on  $\mathbf{m}$  is used, then Thm. 1 only changes by using the different posterior to compute  $\exp(E_q\{\log m_k\})$  and the replacement or removal of (9). We have used the entropic prior here with a large  $\lambda$  as a replacement to the Dirichlet in some experiments to produce quiet different components.

## 5 Experimental Results

Previous researchers have presented results comparing this family of methods with LSI in tasks such as document recall based on the reduced features [2], perplexity evaluation of the dimensionality reduction achieved [2, 4], and comparisons with other statistical models for dimensionality reduction of discrete data [6]. Experiments presented here instead focus on a number of different aspects of the methods highlighted in the analysis. I apply the algorithms to both

bag-of-words document data and word bigram data since these turn out to have very different statistical properties. Note analysis of bigram data should not be viewed as a thesaurus discovery task since this should include part of speech information and word linkage from a dictionary as well.

Bigram data was collected about words from a significant portion of the English language documents in Google’s August 2001 crawl. Identifying sentence breaks is a difficult task in HTML as seemingly random lists of words occur not infrequently in web pages, and space does not allow description of methods here. The bigram data is 17% non-zero for the matrix of the top 5000 words. The top word “to” has 139,597,023 occurrences and the 5,000-th word “charity” has 920,343 occurrences. The most frequent bigram is “to be” with 20,971,200 occurrences, while the 1,000-th most frequent is “included in” at 2,333,447 occurrences. David Lewis’ Reuters-21578 collection of newswires was used as the document data. Words occurring less than 10 times in the entire collection were discarded from the bag-of-words representation, and numbers converted to tokens, leaving 10,697 words as features for approximately 20,000 documents.

The code is 1500 lines of C. Space requirements for runtime are  $O(K * (I + J) + S)$  where  $S$  is the size of the input data and each iteration takes  $O(K * (I + J + S))$ . Thus the computational requirements are comparable to an algorithm for extracting the top  $K$  eigenvectors usually used for PCA. Convergence is maybe 10-30 iterations, depending on the accuracy required, slower than its PCA counterpart. Nevertheless, all experiments reported were run on an old Linux laptop overnight. The code outputs a HTML file for navigating documents/words, components, and their assignments for a document.

Useful diagnostic measures reported on below are as follows:

**Expected words per component (EW/C):** the conditional entropy of the word probability vectors in  $\Omega$  given components raised to power 2.

**Expected components per document (EC/D):** the entropy of the component probability vectors averaged over documents raised to power 2.

**Expected components (EC):** the entropy of the observed component probabilities raised to power 2, should be  $O(K)$ .

The plots in Fig 1 show the change of these values as  $K$  is increased. For the Reuters-21578 data, EC/D stays constant at about 2 (not plotted) while EC continues growing almost linearly. For the Google bigram data, EC/D grows as plotted primarily because the sample sizes are very large. Intense subsampling (thinning out the data by a factor of 1000) eventually makes EC/D stay constant at about 4. Thus newswires/words in both data sets are almost equally distributed across components but typically one newswire only belongs to 2 components, whereas one word (for the Google Bigram data) belongs to several more components depending on the sample size.

Note the efficacy of the priors used for parameters  $\alpha$  of the component Dirichlet can be measured by computing the KL divergence between the expected component proportions (according to  $\alpha$ ) and the observed component proportions. Remarkably, on the Reuters-21578 data with  $K=1300$  components, the KL divergence using a prior sample size of one for  $\alpha$  is only 0.22503 compared to 2.24437

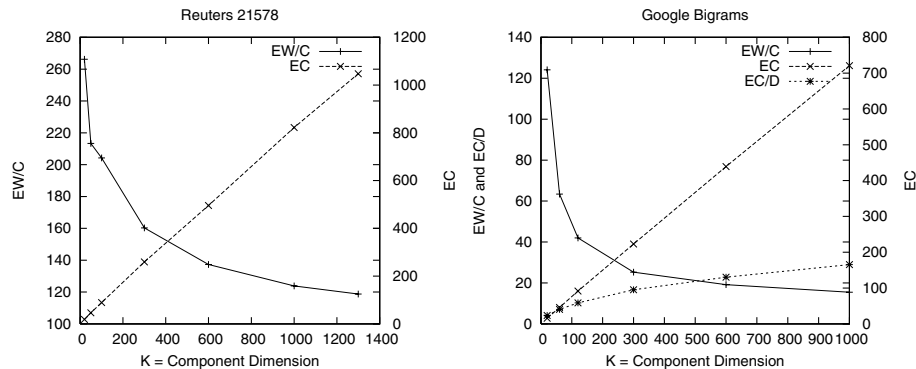


Fig. 1. Diagnostics for the data runs

when the maximum likelihood estimates of Blei *et al.* are used, thus the estimates are on average a factor of four times more accurate with the prior, while other diagnostic measures did remain unchanged. The poor quality of the maximum likelihood estimates is to be expected since they are hierarchical parameters not directly estimated from the data.

The components found from the bigram data vary for different values of  $K$ . For small  $K$  (e.g., 10), general word forms such as verbs, adjectives, etc. are found. As  $K$  increases (20,30,50) these break out into “people verbs”, “internet nouns,” etc. Once  $K$  increases to about 300, the components include things like months, measurements, US states, democracy verbs, emotions, media formats, body parts, and more abstract components such as “aspects of a thing,” “new ideas,” etc. It is this unfolding of components for increasing  $K$  that is most remarkable about the method, and in complete contrast to PCA which simply adds components to the existing top  $K$ .

## 6 Conclusion

The model presented here is a multinomial analogue to PCA of the kind espoused by Collins *et al.* [5], where a multinomial mean is a convex combination of multinomial components (Sct. 2). It is also a model that assigns a single component/topic to each word of a document (Sct. 4.2, also [4]). Thus I support the name *multinomial PCA*: “non-negative matrix factorization” is better descriptive of a non-statistical matrix analysis task, “probabilistic LSI” has a clear comparison with LSI [2] but fails to indicate its far broader applicability, and “latent Dirichlet” focuses on a minor aspect (e.g., replacing the Dirichlet with an entropic prior leaves rewrite rules almost unchanged). The theorems presented here have extended the algorithm of Blei *et al.* [4], simplified their proof, and clarified the relationship with the earlier algorithms, NMF and pLSI. The relationship is rather like k-means clustering versus EM clustering: they use the same generative model but differ in optimization criteria.

Thus while the goal is analogous to PCA, the results of these methods depart radically from PCA and provide a rich source of new opportunities in analyzing discrete data. Finding greater numbers of components does not simply add more components, but rather develops refined components at a completely different scale. Multinomial PCA is therefore ideal for hierarchical analysis. Moreover, current versions of mPCA can have many components per document and it is only due to the small size of the newswires that typically there are 2 components per newswire. In the bigram task, components per word kept increasing beyond 30, highlighting that current mPCA algorithms are really suited for dimensionality reduction and not as a form of relaxed clustering that allows probabilistic assignment across just a few classes.

## Acknowledgements

An earlier theoretical basis for this research was conducted in NASA subcontract NAS2-00065 for the QSS Group in June 2001, and the bigram modeling as a machine learning research consultant for Google, Inc. with Peter Norvig in Sept.Oct. 2001.

## References

- [1] Tipping, M., Bishop, C.: Mixtures of probabilistic principal component analysers. *Neural Computation* **11** (1999) 443–482 [23](#), [24](#)
- [2] Hofmann, T.: Probabilistic latent semantic indexing. In: *Research and Development in Information Retrieval*. (1999) 50–57 [23](#), [31](#), [33](#)
- [3] Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. *Nature* **401** (1999) 788–791 [23](#), [31](#)
- [4] Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. In: *NIPS\*14*. (2002) to appear. [23](#), [28](#), [29](#), [31](#), [33](#)
- [5] Collins, M., Dasgupta, S., Schapire, R.: A generalization of principal component analysis to the exponential family. In: *NIPS\*13*. (2001) [23](#), [24](#), [33](#)
- [6] Hall, K., Hofmann, T.: Learning curved multinomial subfamilies for natural language processing and information retrieval. In: *ICML 2000*. (2000) [23](#), [31](#)
- [7] Sjolander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I. S., Haussler, D.: Dirichlet mixtures: A method for improved detection of weak but significant protein sequence homology. *Computer Applications in the Biosciences* **12** (1996) 327–345 [23](#)
- [8] Brand, M.: Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation* **11** (1999) 1155–1182 [24](#), [28](#)
- [9] Roweis, S.: EM algorithms for PCA and SPCA. In: *NIPS\*10*. (1998) [24](#)
- [10] Ghahramani, Z., Beal, M.: Propagation algorithms for variational Bayesian learning. In: *NIPS*. (2000) 507–513 [25](#), [26](#)
- [11] Buntine, W.: Computation with the exponential family and graphical models. unpublished handouts, NATO Summer School on Graphical Models, Erice, Italy (1996) [25](#)
- [12] Minka, T.: Estimating a Dirichlet distribution. Course notes (2000) [25](#)
- [13] Jordan, M., Ghahramani, Z., Jaakkola, T., Saul, L.: An introduction to variational methods for graphical models. *Machine Learning* **37** (1999) 183–233 [27](#)