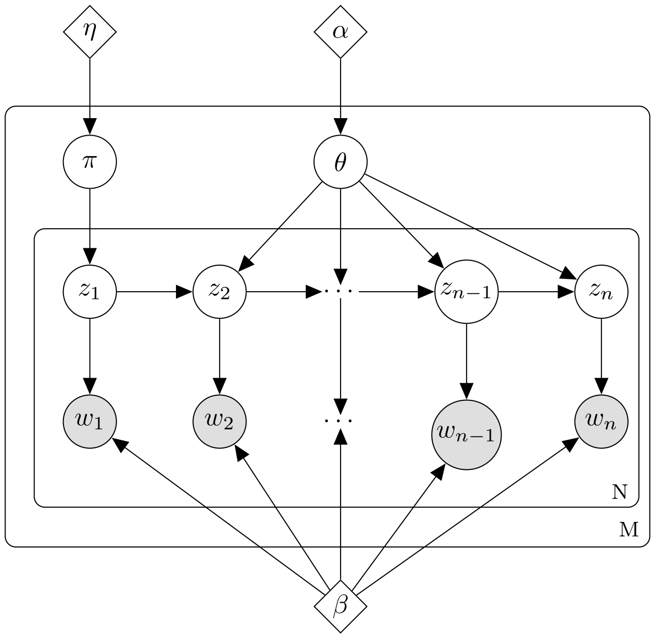
Hidden Markov Topic Model: Variational Bayes Algorithm

Eric Proffitt

June 13, 2016

Notation:

ψ	Digamma function.
diagm$(\cdot)$	Converts a vector to a diagonal matrix.
M	Number of documents.
V	Number of terms in the lexicon.
K	Number of topics.
N_d	Length of document d.



As you can see, I changed your notation a bit. Basically I changed the latent topic choice for each word from x to z, I changed your Dirichlet prior on π from α to η, and I changed the Dirichlet prior on θ from γ to α (since I use γ as one of my variational parameters).

Also, I don't find much improvement in topic quality by putting a prior on β, and it slows down the algorithm considerably, so β remains a hyper-parameter as was done in David Blei's original LDA paper. If for some reason one needs non-zero probabilities across the entire lexicon then I just use Laplace smoothing on β post-fit.

Since you say you're not terribly familiar with VB, this paper doubles as a shotgun treatise on variational Bayes. Our goal is to maximize the marginal log-likelihood

$$\log p(w|\alpha,\eta,\beta) = \sum_{d=1}^{M} \int \int \sum_{z_d} p(z_d,\theta_d,\pi_d \mid w_d,\alpha,\eta,\beta) \log \Big[\frac{p(w_d,z_d,\theta_d,\pi_d \mid \alpha,\eta,\beta)}{p(z_d,\theta_d,\pi_d \mid w_d,\alpha,\eta,\beta)} \Big] \mathrm{d}\theta_d \mathrm{d}\pi_d,$$

where this *canonical* equality follows by Bayes' theorem and a little algebra.

Unfortunately the distribution over latent variables is intractable to compute, thus we approximate it with a simpler distribution, usually by making some extra independence assumptions. This technique is referred to as *mean-field approximation*, and originated in statistical mechanics:

$$\prod_{d=1}^{M} p(z_d,\theta_d,\pi_d \mid w_d,\alpha,\eta,\beta) \approx \prod_{d=1}^{M} q(z_d \mid \phi_d,\lambda_d) q(\theta_d \mid \gamma_d) q(\pi_d \mid \tau_d).$$

$$\begin{aligned}
q(z_{d_1} \mid \lambda_d) &= \mathrm{Cat}(\lambda_d) \\
q(z_{d_n} \mid z_{d_{n-1}}, \phi_d) &= \big[\phi_d\big]_{z_{d_n}, z_{d_{n-1}}} \\
q(\theta_{d_{i=1:K,l}} \mid \gamma_{d_{i=1:K,l}}) &= \mathrm{Dir}(\gamma_{d_{i=1:K,l}}) \\
q(\pi_d \mid \tau_d) &= \mathrm{Dir}(\tau_d)
\end{aligned}$$

It turns out that due to the fact that the KL-divergence is always positive, making this approximation ensures that our new "approximate marginal log-likelihood" lower-bounds the true marginal log-likelihood. Furthermore, due to the continuity of the KL-divergence, as our approximate distribution q gets closer to the true distribution over the latent variables (with the KL-divergence acting as our measure of similarity), our approximate marginal log-likelihood gets closer to the true log-likelihood:

$$\sum_{d=1}^{M}\int\int\sum_{z_d} p(z_d,\theta_d,\pi_d \mid w_d,\alpha,\eta,\beta)\log\Big[\frac{p(w_d,z_d,\theta_d,\pi_d \mid \alpha,\eta,\beta)}{p(z_d,\theta_d,\pi_d \mid w_d,\alpha,\eta,\beta)}\Big]\mathrm{d}\theta_d\mathrm{d}\pi_d$$

$$\geq \sum_{d=1}^{M}\int\int\sum_{z_d} q(z_d \mid \phi_d,\lambda_d)q(\theta_d \mid \gamma_d)q(\pi_d \mid \tau_d)\log\Big[\frac{p(w_d,z_d,\theta_d,\pi_d \mid \alpha,\eta,\beta)}{q(z_d \mid \phi_d,\lambda_d)q(\theta_d \mid \gamma_d)q(\pi_d \mid \tau_d)}\Big]\mathrm{d}\theta_d\mathrm{d}\pi_d$$

$$= \log p(w|\alpha,\eta,\beta) - \sum_{d=1}^{M} D_{\mathrm{KL}}\big(q(z_d \mid \phi_d,\lambda_d)q(\theta_d \mid \gamma_d)q(\pi_d \mid \tau_d) \mid\mid p(z_d,\theta_d,\pi_d \mid w_d,\alpha,\eta,\beta)\big).$$

Note that since the KL-divergence is not symmetric, it's actually the *reverse* KL-divergence that we're using, where q is acting as the true distribution of the underlying random variable. Thus information theorists might take exception, but since we're using the KL-divergence for its properties, and not for it's information theoretic interpretation, we're generally ok with this.

Therefore by iterating back and forth between improving the approximate latent-variable distribution q (by optimizing its own set of parameters ϕ,λ,γ,τ), and optimizing the global hyper-parameters α,η,β, we have turned the problem of fitting our model into an optimization problem.

Using the log function and the linearity property of sums and integrals to break down the above expression, we obtain:

$$\begin{aligned}\sum_{d=1}^{M}\Big(&E_q[\log p(\pi_d \mid \eta)] + \sum_{l=1}^{K} E_q[\log p(\theta_{d_{1:K,l}} \mid \alpha_{1:K,l})]\\ &+ E_q[\log p(z_{d_1} \mid \pi_d)] + \sum_{n=2}^{N_d} E_q[\log p(z_{d_n} \mid z_{d_{n-1}},\theta_d)]\\ &+ \sum_{n=1}^{N_d} E_q[\log p(w_{d_n} \mid z_{d_n},\beta)] - E_q[\log q(\pi_d \mid \tau_d)]\\ &- \sum_{l=1}^{K} E_q[\log q(\theta_{d_{1:K,l}} \mid \gamma_{d_{1:K,l}})] - E_q[\log q(z_{d_1} \mid \lambda_d)]\\ &- \sum_{n=2}^{N_d} E_q[\log q(z_{d_n} \mid z_{d_{n-1}},\phi_d)]\Big).\end{aligned}$$

Next we compute analytic forms for all expectations:

$$\begin{aligned}
L(\lambda,\phi,\tau,\gamma,\eta,\alpha,\beta) = \sum_{d=1}^{M} \Big(& \log\Gamma\big(\sum_{i=1}^{K}\eta_i\big) - \sum_{i=1}^{K}\log\Gamma(\eta_i) + \sum_{i=1}^{K}(\eta_i-1)(\psi(\tau_{d_i}) - \psi\big(\sum_{p=1}^{K}\tau_{d_p}\big)) \\
& + \sum_{l=1}^{K}\big(\log\Gamma\big(\sum_{i=1}^{K}\alpha_{il}\big) - \sum_{i=1}^{K}\log\Gamma(\alpha_{il}) + \sum_{i=1}^{K}(\alpha_{il}-1)(\psi(\gamma_{d_{il}}) - \psi\big(\sum_{p=1}^{K}\gamma_{d_{pl}}\big))\big) \\
& + \sum_{i=1}^{K}\lambda_{d_i}(\psi(\tau_{d_i}) - \psi\big(\sum_{p=1}^{K}\tau_{d_p}\big)) \\
& + \sum_{l=1}^{K}\big(\sum_{n=2}^{N_d} e_l^T\phi_d^{n-2}\lambda_d\big)\big(-\psi\big(\sum_{p=1}^{K}\gamma_{d_{pl}}\big) + \sum_{i=1}^{K}\phi_{d_{il}}\psi(\gamma_{d_{il}})\big) \\
& + \sum_{n=1}^{N_d}(\log\beta_{1:K,w_{d_n}})^T\phi_d^{n-1}\lambda_d \\
& - \big(\log\Gamma\big(\sum_{i=1}^{K}\tau_{d_i}\big) - \sum_{i=1}^{K}\log\Gamma(\tau_{d_i}) + \sum_{i=1}^{K}(\tau_{d_i}-1)(\psi(\tau_{d_i}) - \psi\big(\sum_{p=1}^{K}\tau_{d_p}\big))\big) \\
& - \sum_{l=1}^{K}\big(\log\Gamma\big(\sum_{i=1}^{K}\gamma_{d_{il}}\big) - \sum_{i=1}^{K}\log\Gamma(\gamma_{d_{il}}) + \sum_{i=1}^{K}(\gamma_{d_{il}}-1)(\psi(\gamma_{d_{il}} - \psi\big(\sum_{p=1}^{K}\gamma_{d_{pl}}\big))\big) \\
& - \sum_{i=1}^{K}\lambda_{d_i}\log\lambda_{d_i} - \sum_{l=1}^{K}\big(\sum_{n=2}^{N_d} e_l^T\phi_d^{n-2}\lambda_d\big)\big(\sum_{i=1}^{K}\phi_{d_{il}}\log\phi_{d_{il}}\big)\Big).
\end{aligned}$$

Thus our goal is now to maximize (via coordinate ascent) our objective function L:

$$\max_{\lambda,\phi,\tau,\gamma,\eta,\alpha,\beta} L(\lambda,\phi,\tau,\gamma,\eta,\alpha,\beta).$$

Setting partial derivatives equal to zero and analytically solving for optimal points suffices for parameters τ,γ and β:

$$\begin{aligned}
\tau_{d_i} &= \eta_i + \lambda_{d_i} \\
\gamma_{d_{il}} &= \alpha_{il} + \phi_{d_{il}} \cdot \big[\sum_{n=2}^{N_d}\phi_d^{n-2}\lambda_d\big]_l \\
\hat{\beta}_j &= \sum_{d=1}^{M}\sum_{n=1}^{N_d} w_n^j\phi_d^{n-1}\lambda_d \\
\beta_{ij} &= \hat{\beta}_{ij}/\sum_{v=1}^{V}\hat{\beta}_{iv}
\end{aligned}$$

For optimizing parameters α and η we can use an interior point Newton method, the computations for their respective gradients and inverse Hessians proceed in a fashion very similar to that done for α in David Blei's original LDA paper (found on his website).

For λ we use Lagrange multipliers to form the objective function

$$\begin{aligned}
L_{\lambda_d} = \Big[&\big[\psi(\tau_{d_l}) - \psi\big(\sum_{p=1}^{K} \tau_{d_p}\big)\big]_{l=1:K}^{T} \\
&+ \big[\big(-\psi\big(\sum_{p=1}^{K} \gamma_{d_{pl}}\big) + \sum_{i=1}^{K} \phi_{d_{il}} \psi(\gamma_{d_{il}})\big) e_l^T \sum_{n=2}^{N_d} \phi_d^{n-2}\big]_{l=1:K}^{T} \\
&+ \big(\sum_{n=1}^{N_d} (\log \beta_{1:K, w_{d_n}})^T \phi_d^{n-1} \big) \\
&- \big[\big(\sum_{i=1}^{K} \phi_{d_{il}} \log \phi_{d_{il}}\big) e_l^T \sum_{n=2}^{N_d} \phi_d^{n-2}\big]_{l=1:K}^{T} \Big] \cdot \lambda_d \\
&- \sum_{i=1}^{K} \lambda_{d_i} \log \lambda_{d_i} + v_d(\mathbf{1}^T \lambda_d - 1),
\end{aligned}$$

where the final piece involving v_d is the Lagrange multiplier ensuring that λ_d remains a unit vector. The piece inside the large square brackets is a length K vector, denoting this vector by C_d, and then taking the derivative with respect to λ_d, we obtain:

$$\begin{aligned}
L_{\lambda_d} &= C_d^T \lambda_d - \sum_{i=1}^{K} \lambda_{d_i} \log \lambda_{d_i} + v_d(\mathbf{1}^T \lambda_d - 1). \\
\frac{\partial L_{\lambda_d}}{\partial \lambda_d} &= C_d^T - (\log \lambda_d)^T - \mathbf{1}^T + v_d \mathbf{1}^T = 0
\end{aligned}$$

Solving for λ_d, we obtain:

$$\lambda_d = \exp(C_d + v_d \mathbf{1} - \mathbf{1}),$$

note that exp is acting coordinate-wise on the vector $C_d + v_d \mathbf{1} - \mathbf{1}$. Now substituting this value back in to L_{λ_d} and taking the derivative with respect to v_d, we have:

$$\begin{aligned}
G_{v_d} = C_d^T \exp(C_d + v_d \mathbf{1} - \mathbf{1}) &- (C_d + v_d \mathbf{1} - \mathbf{1}) \exp(C_d + v_d \mathbf{1} - \mathbf{1}) \\
&+ v_d \big(\mathbf{1}^T \exp(C_d + v_d \mathbf{1} - \mathbf{1}) - 1\big)
\end{aligned}$$

$$\frac{\partial G_{v_d}}{\partial v_d} = \mathbf{1^T} \exp(C_d + v_d \mathbf{1} - \mathbf{1}) - 1 = 0.$$

This equation can then be solved for v_d using Newton's method.

Finally, we come to the problem of optimizing ϕ. The objective function for ϕ, for a particular document d, is given by:

$$\begin{aligned}L_{\phi_d} = &\Big[-\psi\big(\sum_{p=1}^{K}\gamma_{d_{pl}}\big) + \sum_{i=1}^{K}\phi_{d_{il}}\big(\psi(\gamma_{d_{il}}) - \log\phi_{d_{il}}\big)\Big]_{l=1:k}\cdot\big(\sum_{n=2}^{N_d}\phi_d^{n-2}\big)\cdot\lambda_d\\ &+\sum_{n=1}^{N_d}[\log\beta_{1:K,w_{d_n}}]^T\cdot\phi_d^{n-1}\cdot\lambda_d\end{aligned}$$

$$\begin{aligned}\frac{\partial L_{\phi_d}}{\partial\phi_d} = &[\psi(\gamma_{d_{il}}) - \log\phi_{d_{il}} - 1]_{i=1:K,\ l=1:K}\cdot\text{diagm}\Big(\big(\sum_{n=2}^{N_d}\phi_d^{n-2}\big)\cdot\lambda_d\Big)\\ &+\sum_{n=3}^{N_d}\sum_{r=0}^{n-3}\Big((\phi_d^r)^T\cdot\Big[-\psi\big(\sum_{i=1}^{K}\gamma_{d_{ij}}\big) + \sum_{i=1}^{K}\phi_{d_{il}}\big(\psi(\gamma_{d_{ij}}) - \log\gamma_{d_{ij}}\big)\Big]^T_{j=1:K}\cdot\lambda_d^T\cdot(\phi_d^{n-3-r})^T\Big)\\ &+\sum_{n=2}^{N_d}\sum_{r=0}^{n-2}(\phi_d^r)^T\cdot[\log\beta_{in}]_{i=1:K}\cdot\lambda_d^T\cdot(\phi_d^{n-2-r})^T.\end{aligned}$$

We have two important features of this optimization problem working in our favor. First, the problem is low-dimensional, since ϕ_d is a matrix of size $K\times K$, where K is the number of topics in our topic model, which is almost always < 100. Second, the gradient, as seen above, is analytically computable (just barely, see formula (91) in the Matrix Cookbook).

Unfortunately, there are also a number of features which work against us. First off, the optimization problem is constrained, ϕ_d is a left stochastic matrix (all entries nonnegative with each column summing to 1). Thus we have the (fortunately linear) constraints

$$\begin{aligned}&\phi_d^T\mathbf{1} = \mathbf{1}\\ &\phi_d \geq 0.\end{aligned}$$

However a possibly even more serious problem is the complexity of the gradient. Each evaluation of the gradient involves hundreds of matrix multiplications, which means large step-number optimizations such as gradient ascent are prohibitively expensive. If ϕ was a global hyper-parameter which only needed to be optimized once for every pass through the corpus, then this might be ok, but ϕ is a document-level variational parameter, which means we have a unique ϕ_d attached to each document, meaning that any optimization procedure needs to be *fast*, on the order of a few milliseconds per document.

I've tried a number of optimization methods including projected gradient ascent along the manifold of stochastic matrices, various gradient ascent and gradient-free methods, numerical hessian, numerical gradient and numerical hessian. This was several months ago so I can't exactly remember why these methods all failed, most I think were too slow or the augmentation required to keep it on the surface of stochastic matrices caused too much error and it didn't optimize at all, but I'd have to try some of them again to be sure. There might be a way to leverage some statistical algorithm like the forward-backwards algorithm, I tried one implementation of this for general HMMs in Julia, but it was super slow.

I have everything else coded up and working. Basically if I just don't optimize the ϕ variables and leave them fixed, then the remaining variables optimize fine and at a good speed. Also beware I wrote these notes up fairly quickly so there might be some errors in the math, thus don't hesitate to ask if you have questions or if you find what you think is an error.