Variational Bayes Derivation of Latent Dirichlet Allocation

Simple LDA, not smoothed LDA

September 8, 2016

Contents

1 Model

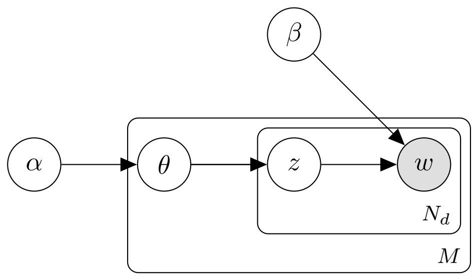

Figure 1: LDA Model

Variables:

- M: a number of document
- N_d: a number of words in document d
- $w_{d,i}$: a word
- β: (number of topic K) $\times$ (number of unique words V)
  - $\beta_{k,v} = p(w_{d,i} = v | z_{d,i} = k)$ is the probability of the word v occurring given the topic k

– Caution: At least in the Python code, β is $V \times K$ (maybe in C as well)

- $z_{d,i}$: latent topic

2 Derivation with Code

2.1 Evidence lower bound

$$
\begin{aligned}
\log p(\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\beta}) &= \log \int \sum_{\boldsymbol{z}} p(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\theta}|\boldsymbol{\alpha},\boldsymbol{\beta}) d\boldsymbol{\theta} \qquad \text{(intdotuce latent variables)} && (1)\\
&= \log \int \sum_{\boldsymbol{w}} q(\boldsymbol{z},\boldsymbol{\theta}) \frac{p(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\theta}|\boldsymbol{\alpha},\boldsymbol{\beta})}{q(\boldsymbol{z},\boldsymbol{\theta})} d\boldsymbol{\theta} && (2)\\
&\leq \int \sum_{\boldsymbol{z}} q(\boldsymbol{z},\boldsymbol{\theta}) \log \frac{p(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\theta}|\boldsymbol{\alpha},\boldsymbol{\beta})}{q(\boldsymbol{z},\boldsymbol{\theta})} d\boldsymbol{\theta} \qquad \because \text{Jensen's Inequality} && (3)\\
&\equiv F[q(\boldsymbol{z},\boldsymbol{\theta})] && (4)
\end{aligned}
$$

From factorization assumption:

$$
q(\boldsymbol{z},\boldsymbol{\theta}) = \left[\prod_{d=1}^{M}\prod_{i=1}^{N_d} q(z_{d,i})\right]\left[\prod_{d=1}^{M} q(\boldsymbol{\theta}_d)\right] \tag{5}
$$

Expand the joint distribution using Bayes' Theorem:

$$
\begin{aligned}
p(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\theta}|\boldsymbol{\alpha},\boldsymbol{\beta}) &= p(\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{\theta}) \quad \underbrace{p(\boldsymbol{z},\boldsymbol{\theta}|\boldsymbol{\alpha},\boldsymbol{\beta})}_{p(\boldsymbol{z}|\boldsymbol{\theta},\boldsymbol{\alpha},\boldsymbol{\beta})p(\boldsymbol{\theta}|\boldsymbol{\alpha},\boldsymbol{\beta})} && (6)\\
&= p(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\beta})p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \qquad \because \text{Graphical Model} && (7)\\
&= \left[\prod_{d=1}^{M}\prod_{i=1}^{N_d} p(w_{d,i}|\beta_{z_{d,i}})p(z_{d,i}|\boldsymbol{\theta}_d)\right]\left[\prod_{d=1}^{M} p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\right] && (8)
\end{aligned}
$$

Evidence lower bound (ELBO) is:

$$
\begin{aligned}
F[q(\boldsymbol{z},\boldsymbol{\theta})] &= \int \sum_{\boldsymbol{z}} q(\boldsymbol{z})q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\beta})p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{q(\boldsymbol{z})q(\boldsymbol{\theta})} d\boldsymbol{\theta} && (9)\\
&= \int \sum_{\boldsymbol{z}} q(\boldsymbol{z})q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\beta})p(\boldsymbol{z}|\boldsymbol{\theta})}{q(\boldsymbol{z})} d\boldsymbol{\theta} + \int \sum_{\boldsymbol{z}} q(\boldsymbol{z})q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} && (10)\\
&= \int \sum_{\boldsymbol{z}} q(\boldsymbol{z})q(\boldsymbol{\theta}) \log p(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\beta})p(\boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{\theta} - \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log q(\boldsymbol{z}) + \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} && (11)\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(integrate out unrelated variables)}\\
&= \int \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i})q(\boldsymbol{\theta}_d) \log p(w_{d,i}|z_{d,i},\beta)p(z_{d,i}|\boldsymbol{\theta}_d) d\boldsymbol{\theta}_d\\
&\quad - \sum_{d=1}^{M}\sum_{i=1}^{N_d}\sum_{k=1}^{K} q(z_{d,i}=k) \log q(z_{d,i}=k) && (12)\\
&\quad - \sum_{d=1}^{M} \underbrace{\int q(\boldsymbol{\theta}_d) \log \frac{q(\boldsymbol{\theta}_d)}{p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})} d\boldsymbol{\theta}_d}_{\mathrm{KL}[q(\boldsymbol{\theta}_d)||p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})]}
\end{aligned}
$$

2.2 Update Equation of $q(z_{d,i})$

2.2.1 Derivation

$$\widetilde{F}[q(z_{d,i})] = \sum_{k=1}^{K} q(z_{d,i}=k) \int q(\boldsymbol{\theta}_d) \log\{p(w_{d,i}|z_{d,i},\beta_{k,i})p(z_{d,i}=k|\boldsymbol{\theta}_d)\} d\boldsymbol{\theta}_d - \sum_{k=1}^{K} q(z_{d,i}=k) \log q(z_{d,i}=k) \tag{13}$$

Use variational inference:

$$\frac{\delta \widetilde{F}[q(z_{d,i})]}{\delta q(z_{d,i}=k)} = \frac{\partial \widetilde{F}[q(z_{d,i})]}{\partial q(z_{d,i}=k)} = \int q(\boldsymbol{\theta}_d) \log(\beta_{k,w_{d,i}}, \theta_{d,k}) d\boldsymbol{\theta}_d - \log q(z_{d,i}=k) - 1 = 0 \tag{14}$$

Hence,

$$\begin{aligned}
q(z_{d,i}=k) &\propto \exp\left[\int q(\boldsymbol{\theta}_d) \log(\beta_{k,w_{d,i}}\theta_{d,k}) d\boldsymbol{\theta}_d\right] && (15)\\
&= \exp\left[\int q(\boldsymbol{\theta}_d) \log(\beta_{k,w_{d,i}}) d\boldsymbol{\theta}_d\right] \exp\left[\int q(\boldsymbol{\theta}_d) \log(\theta_{d,k}) d\boldsymbol{\theta}_d\right] && (16)\\
&= \beta_{k,w_{d,i}} \ \exp\left[\mathbb{E}_{q(\boldsymbol{\theta}_d)} \log(\theta_{d,k})\right] && (17)\\
&\propto \beta_{k,w_{d,i}} \frac{\exp\left[\Psi(\xi^{\theta}_{d,k})\right]}{\exp\left[\Psi(\sum_{k'=1}^{K} \xi^{\theta}_{d,k'})\right]} \qquad \xi^{\theta}_{d,k} = \mathbb{E}_{q(\boldsymbol{z}_d)}[N_{d,k}] + \alpha_k && (18)
\end{aligned}$$

Note $\Psi(\ )$ is a digamma function. If $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$ is a K-dimensional Dirichlet distribution,

$$\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{\alpha})}[\log \theta_k] = \Psi(\alpha_k) - \Psi\left(\sum_{k=1}^{K} \alpha_k\right)$$

2.2.2 Code

Caution: At least in the Python code, β is $V \times K$ (maybe in C as well). Probably normalization comes later.

In Python,

```
q = lda.mnormalize(matrix(beta[d[0],:]) * matrix(diag(exp(digamma(alpha0 +
    nt))[0])), 1)
```

In C,

```
/* In vbem.c */
/* vb-estep */
for (k = 0; k < K; k++)
        ap[k] = exp(psi(alpha[k] + nt[k]));
/* accumulate q */
for (l = 0; l < L; l++)
        for (k = 0; k < K; k++)
                q[l][k] = beta[d->id[l]][k] * ap[k];
/* normalize q */
for (l = 0; l < L; l++) {
        z = 0;
        for (k = 0; k < K; k++)
                z += q[l][k];
        for (k = 0; k < K; k++)
                q[l][k] /= z;
```

```
}
```

`ap[k]` is (the numerator of) the second term in (18).

2.3 Update Equation of $q(\boldsymbol{\theta}_d)$

2.3.1 Derivation

Again, ELBO is

$$
\begin{aligned}
F[q(\boldsymbol{z},\boldsymbol{\theta})] = \int \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i})q(\boldsymbol{\theta}_d)\log p(w_{d,i}|z_{d,i},\beta)p(z_{d,i}|\theta_d)d\boldsymbol{\theta}_d - \sum_{d=1}^{M}\sum_{i=1}^{N_d}\sum_{k=1}^{K} q(z_{d,i}=k)\log q(z_{d,i}=k) \\
- \sum_{d=1}^{M} \underbrace{\int q(\boldsymbol{\theta}_d)\log\frac{q(\boldsymbol{\theta}_d)}{p(\boldsymbol{\theta}_d)|\boldsymbol{\alpha})}d\boldsymbol{\theta}_d}_{\mathrm{KL}[q(\boldsymbol{\theta}_d)||p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})]}
\end{aligned}
\tag{19}
$$

We use terms only related to $\boldsymbol{\theta}$.

$$
\widetilde{F}[q(\boldsymbol{\theta})] = \int q(\boldsymbol{\theta})\sum_{\boldsymbol{z}} q(\boldsymbol{z})\log p(\boldsymbol{z}|\boldsymbol{\theta})d\boldsymbol{\theta} - \int q(\boldsymbol{\theta})\log\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{\alpha})}d\boldsymbol{\theta} \tag{20}
$$

$$
\widetilde{F}[q(\boldsymbol{\theta}_d)] = \int q(\boldsymbol{\theta}_d)\sum_{\boldsymbol{z}} q(\boldsymbol{z})\sum_{i=1}^{N_d}\log p(z_{d,i}|\boldsymbol{\theta}_d)d\boldsymbol{\theta}_d - \int q(\boldsymbol{\theta}_d)\log\frac{q(\boldsymbol{\theta}_d)}{p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})}d\boldsymbol{\theta}_d \tag{21}
$$

Using variational inference,

$$
\frac{\delta\widetilde{F}[q(\boldsymbol{\theta}_d)]}{\delta q(\boldsymbol{\theta}_d)} = \frac{\partial\widetilde{F}[q(\boldsymbol{\theta}_d)]}{\partial q(\boldsymbol{\theta}_d)} = \sum_{\boldsymbol{z}} q(\boldsymbol{z})\sum_{i=1}^{N_d}\log p(z_{d,i}|\boldsymbol{\theta}_d) - \log\frac{q(\boldsymbol{\theta}_d)}{p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})} - 1 = 0 \tag{22}
$$

Before we move on, let's check some deformations:

- Dirichlet distribution

$$
\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \equiv \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\pi_k^{\alpha_k-1} \tag{23}
$$

- If we consider a category k in a document d, the average number of words that belong to the category k under certain latent variables is

$$
\mathbb{E}_{q(\boldsymbol{z}_d)}[N_{d,k}] = \sum_{i=1}^{N_d} q(z_{d,i}=k) \tag{24}
$$

- Remember $z_{d,i} \sim \mathrm{Multi}(\boldsymbol{\theta}_d)$ (Sato pp.26-27, Equation 2.1). Be careful that a word belongs to a category or not, so we can use $\delta(z_{d,i}=k)$ here.

$$
p(z_{d,i}|\boldsymbol{\theta}_d) = \prod_{k=1}^{K}\theta_{d,k}^{\delta(z_{d,i}=k)} \tag{25}
$$

Now, we can back to the variational inference

$$
\begin{aligned}
q(\boldsymbol{\theta}_d) &\propto p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \exp\left[\sum_{\boldsymbol{z}} q(\boldsymbol{z}) \sum_{i=1}^{N_d} \log p(z_{d,i}|\boldsymbol{\theta}_d)\right] && (26)\\
&\propto \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k-1} \exp\left[\sum_{\boldsymbol{z}} q(\boldsymbol{z}) \sum_{i=1}^{N_d}\sum_{k=1}^{K} \delta(z_{d,i}=k)\log\theta_{d,k}\right] && (27)\\
&= \exp\left[\sum_{k=1}^{K}(\alpha_k-1)\log\theta_{d,k}\right] \exp\left[\sum_{k=1}^{K}\sum_{i=1}^{N_d} q(z_{d,i}=k)\log\theta_{d,k}\right] && (28)\\
&= \exp\left[\sum_{k=1}^{K}(\alpha_k-1)\log\theta_{d,k}\right] \exp\left[\sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{z}_d)}[N_{d,k}]\log\theta_{d,k}\right] && (29)\\
&= \exp\left[\sum_{k=1}^{K}(\mathbb{E}_{q(\boldsymbol{z}_d)}[N_{d,k}]+\alpha_k-1)\log\theta_{d,k}\right] && (30)\\
&= \prod_{k=1}^{K} \theta_{d,k}^{\mathbb{E}_{q(\boldsymbol{z}_d)}[N_{d,k}]+\alpha_k-1} && (31)
\end{aligned}
$$

From (27) to (28), we marginalize the equation with respect to $q(\boldsymbol{z})$. For various $z_{d,i}$ in $\boldsymbol{z}$, some take $\delta(z_{d,i}=k)=0$ and other take $\delta(z_{d,i}=k)=1$. We only need to consider those that are $\delta(z_{d,i}=k)=1$.

If we define $\xi^{\theta}_{d,k}=\mathbb{E}_{q(\boldsymbol{z}_d)}[N_{d,k}]+\alpha_k$, $q(\boldsymbol{\theta}_d)$ is a Dirichlet distribution whose parameters are $\boldsymbol{\xi}^{\theta}_d=(\xi^{\theta}_{d,1},\xi^{\theta}_{d,2},\cdots,\xi^{\theta}_{d,K})$. We can easily normalize it:

$$
q(\boldsymbol{\theta}_d|\boldsymbol{\xi}^{\theta}_d)=\frac{\Gamma(\sum_{k=1}^{K}\xi^{\theta}_{d,k})}{\prod_{k=1}^{K}\Gamma(\xi^{\theta}_{d,k})}\prod_{k=1}^{K}\theta_{d,k}^{\xi^{\theta}_{d,k}-1} \tag{32}
$$

2.3.2 Code

2.3.2.1 $\mathbb{E}_{q(\boldsymbol{z}_d)}[N_{d,k}]$ in Equation (24)

In the dataset, we only have how many times each word appears, so we calculate (number of times a word appears) $\times\ q(z_{d,i}=k)$

In Python,

```
nt = matrix(di[1]) * q
```

In C,

```
/* In vbem.c */
for (k = 0; k < K; k++) {
        z = 0;
        for (l = 0; l < L; l++)
                z += q[l][k] * d->cnt[l];
        nt[k] = z;
}
```

What are stored in `di[1]` and `d->cnt[1]` is the word count for each word in each document.

2.3.2.2 $\boldsymbol{\xi}^{\theta}_d$ in Equation (32)

We only need parameters of Dirichlet distribution. In Python,

```
alpha = alpha0 + nt
  # corresponds to Sato Eq (3.89)
  # for all k in d
```

2.4 Update Equation of β

2.4.1 Derivation

From Equation (12), extract parts related to β,

$$\widetilde{F}[\beta] = \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i}) \log p(w_{d,i}|z_{d,i},\beta)p(z_{d,i}) \tag{33}$$

$$= \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i}) \log \left(\prod_{k=1}^{K} \beta_{z_{d,i}=k}^{N_{d,i}} \right) p(z_{d,i}) \tag{34}$$

$$= \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i}) \sum_{k=1}^{K} \log \left(\beta_{z_{d,i}=k}^{N_{d,i}} \right) p(z_{d,i}) \tag{35}$$

$$= \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i}) \sum_{k=1}^{K} \{ N_{d,i} \log \left(\beta_{z_{d,i}=k} \right) + \log p(z_{d,i}) \} \tag{36}$$

Here, I used the fact that $w_{d,i} \sim \mathrm{Multi}(z_{d,i},\beta)$. Certain value s appearing n_s times in n times try of Multinomial distribution is (Sato p.27)

$$\frac{n!}{\prod_{s=1}^{S} n_s!} \prod_{s=1}^{S} \pi_s^{n_s} \tag{37}$$

In codes, we only know the total number of a word appearance, that is $N_{d,i}$ for a word ID i in document d. Since $\beta_{k,v}$ is the probability of the word v occurring in the topic k, we can use the formula in Equation (37). Hence,

$$\frac{\partial \widetilde{F}[\beta]}{\partial \beta} = \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i}) \sum_{k=1}^{K} \frac{N_{d,i}}{\beta_{z_{d,i}=k}} - 1 \tag{38}$$

$$= \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i}) N_{d,i} \sum_{k=1}^{K} \frac{1}{\beta_{z_{d,i}=k}} - 1 = 0 \tag{39}$$

For fixed k,

$$\beta_{z_{d,i}=k} = \beta_{k,i} \propto \sum_{d=1}^{M}\sum_{i=1}^{N_d} q(z_{d,i}) N_{d,i} \tag{40}$$

2.4.2 Code: `accum_beta()`

This makes matrix β, (number of topic K) $\times$ (number of unique words V)[†]. At least in the Python code, β is $V \times K$ (maybe in C as well). It corresponds to §5.3 and §A.4.1 in the original article. Variational EM algorithm is used.

In Python (mpre details are in `LDA-explanation.ipynb`),

```
def accum_beta(betas, q, t):
  # t = d[i]
  betas[t[0],:] += matrix(diag(t[1])) * q
  return betas
```

Matrix `betas[t[0],:]` is (number of unique words in a document) $\times$ (number of class (category)). `t[0]` is word IDs.

In C,

[†] $\beta_{k,v} = p(w_{d,i} = v | z_{d,i} = k)$ is the probability of the word v occurring given the topic k.

```c
/* in learn.c */
int i, k;
int n = dp->len;

for (i = 0; i < n; i++)
        for (k = 0; k < K; k++)
                betas[dp->id[i]][k] += q[i][k] * dp->cnt[i];
```

Original article says

$$\beta_{ij} \propto \sum_{d=1}^{M} \sum_{n=1}^{Nd} \phi_{dni}^{*} w_{dn}^{j} \tag{41}$$

for this part. ϕ_{dni} in the original article (n is the nth word in the document d, i is the topic index. should be denoted as ϕ_{dik} in this note) might correspond to q in the codes, which is $q(z_{d,i} = k)$ (here, i is the ith word in the document d). In the original article, each word is counted up, so $\sum_{n=1}^{N_d} w_{dn}^{j}$ is the total number of word j that comes from topic i in document d (original article Eq.(9), p.1006), which is summed in advance in dataset (`t[1]`, `dp->cnt[i]`).

There is a loop in the code, so part of the β is updated every time when looping over the all documents (be careful again that β is $V \times K$ in codes, which is different from the original paper).

```python
for i in range(n): # n is the number of documents
  gamma,q = lda.vbem(d[i], beta, alpha, demmax)
  gammas[i,:] = gamma # gamma = xi_d ? cf. eq(32)
  betas = lda.accum_beta(betas,q,d[i])
```

Reference

1. Blei et al., ”Latent Dirichlet Allocation”, The Journal of Machine Learning Research, 2003.

2. Mochihashi, Daichi. ”lda, a Latent Dirichlet Allocation package” at http://chasen.org/ daiti-m/dist/lda/ for C code

3. Sato, Makoto. 「PythonでLDAを実装してみる」at http://satomacoto.blogspot.jp/2009/12/pythonlda.html for Python code