

Variational Bayes Derivation of Latent Dirichlet Allocation

Simple LDA, not smoothed LDA

September 8, 2016

1 Model

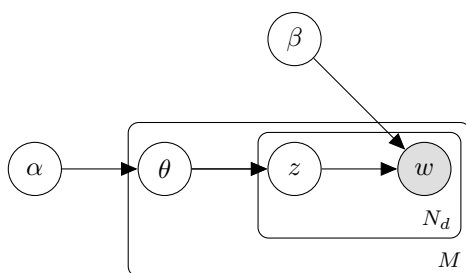


Figure 1: LDA Model

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

$$\log p(\mathbf{w} | \alpha, \beta) = \log \int \sum_{\mathbf{z}} p(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta} | \alpha, \beta) d\boldsymbol{\theta} \quad (\text{introduce latent variables}) \quad (1)$$

$$= \log \int \sum_{\mathbf{w}} q(\mathbf{z}, \boldsymbol{\theta}) \frac{p(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta} | \alpha, \beta)}{q(\mathbf{z}, \boldsymbol{\theta})} d\boldsymbol{\theta} \quad (2)$$

$$\leq \int \sum_{\mathbf{z}} q(\mathbf{z}, \boldsymbol{\theta}) \log \frac{p(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta} | \alpha, \beta)}{q(\mathbf{z}, \boldsymbol{\theta})} d\boldsymbol{\theta} \quad \because \text{Jensen's Inequality} \quad (3)$$

$$\equiv F[q(\mathbf{z}, \boldsymbol{\theta})] \quad (4)$$

From factorization assumption:

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

Expand the joint distribution using Bayes' Theorem:

$$p(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\mathbf{w} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{z}, \boldsymbol{\theta}) \underbrace{p(\mathbf{z}, \boldsymbol{\theta} | \boldsymbol{\alpha}, \boldsymbol{\beta})}_{p(\mathbf{z} | \boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\boldsymbol{\theta} | \boldsymbol{\alpha}, \boldsymbol{\beta})} \quad (6)$$

$$= p(\mathbf{w} | \mathbf{z}, \boldsymbol{\beta}) p(\mathbf{z} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | \boldsymbol{\alpha}) \quad \because \text{Graphical Model} \quad (7)$$

$$= \left[\prod_{d=1}^M \prod_{i=1}^{N_d} p(w_{d,i} | \beta_{z_{d,i}}) p(z_{d,i} | \boldsymbol{\theta}_d) \right] \left[\prod_{d=1}^M p(\boldsymbol{\theta}_d | \boldsymbol{\alpha}) \right] \quad (8)$$

Evidence lower bound (ELBO) is:

$$F[q(\mathbf{z}, \boldsymbol{\theta})] = \int \sum_{\mathbf{z}} q(\mathbf{z}) q(\boldsymbol{\theta}) \log \frac{p(\mathbf{w} | \mathbf{z}, \boldsymbol{\beta}) p(\mathbf{z} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | \boldsymbol{\alpha})}{q(\mathbf{z}) q(\boldsymbol{\theta})} d\boldsymbol{\theta} \quad (9)$$

$$= \int \sum_{\mathbf{z}} q(\mathbf{z}) q(\boldsymbol{\theta}) \log \frac{p(\mathbf{w} | \mathbf{z}, \boldsymbol{\beta}) p(\mathbf{z} | \boldsymbol{\theta})}{q(\mathbf{z})} d\boldsymbol{\theta} + \int \sum_{\mathbf{z}} q(\mathbf{z}) q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta} | \boldsymbol{\alpha})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} \quad (10)$$

$$= \int \sum_{\mathbf{z}} q(\mathbf{z}) q(\boldsymbol{\theta}) \log p(\mathbf{w} | \mathbf{z}, \boldsymbol{\beta}) p(\mathbf{z} | \boldsymbol{\theta}) d\boldsymbol{\theta} - \sum_{\mathbf{z}} q(\mathbf{z}) \log q(\mathbf{z}) + \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta} | \boldsymbol{\alpha})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} \quad (11)$$

(integrate out unrelated variables)

$$\begin{aligned} &= \int \sum_{d=1}^M \sum_{i=1}^{N_d} q(z_{d,i}) q(\boldsymbol{\theta}_d) \log p(w_{d,i} | z_{d,i}, \beta) p(z_{d,i} | \boldsymbol{\theta}_d) d\boldsymbol{\theta}_d \\ &\quad - \sum_{d=1}^M \sum_{i=1}^{N_d} \sum_{k=1}^K q(z_{d,i} = k) \log q(z_{d,i} = k) \\ &\quad - \sum_{d=1}^M \underbrace{\int q(\boldsymbol{\theta}_d) \log \frac{q(\boldsymbol{\theta}_d)}{p(\boldsymbol{\theta}_d | \boldsymbol{\alpha})} d\boldsymbol{\theta}_d}_{\text{KL}[q(\boldsymbol{\theta}_d) || p(\boldsymbol{\theta}_d | \boldsymbol{\alpha})]} \end{aligned} \quad (12)$$

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

2.2.1 Derivation

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

Use variational inference:

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

Hence,

$$q(z_{d,i} = k) \propto \exp \left[\int q(\boldsymbol{\theta}_d) \log(\beta_{k,w_{d,i}} \theta_{d,k}) d\boldsymbol{\theta}_d \right] \quad (15)$$

$$= \exp \left[\int q(\boldsymbol{\theta}_d) \log(\beta_{k,w_{d,i}}) d\boldsymbol{\theta}_d \right] \exp \left[\int q(\boldsymbol{\theta}_d) \log(\theta_{d,k}) d\boldsymbol{\theta}_d \right] \quad (16)$$

$$= \beta_{k,w_{d,i}} \exp \left[\mathbb{E}_{q(\boldsymbol{\theta}_d)} \log(\theta_{d,k}) \right] \quad (17)$$

$$\propto \beta_{k,w_{d,i}} \frac{\exp \left[\Psi(\xi_{d,k}^\theta) \right]}{\exp \left[\Psi \left(\sum_{k'=1}^K \xi_{d,k'}^\theta \right) \right]} \quad \xi_{d,k}^\theta = \mathbb{E}_{q(\mathbf{z}_d)} [N_{d,k}] + \alpha_k \quad (18)$$

Note $\Psi(\cdot)$ 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)

In Python,

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

In C,

```
1 /* In vbem.c */
2 /* vb-estep */
3 for (k = 0; k < K; k++)
4     ap[k] = exp(psi(alpha[k] + nt[k]));
5 /* accumulate q */
6 for (l = 0; l < L; l++)
7     for (k = 0; k < K; k++)
8         q[l][k] = beta[d->id[l]][k] * ap[k];
9 /* normalize q */
10 for (l = 0; l < L; l++) {
11     z = 0;
12     for (k = 0; k < K; k++)
13         z += q[l][k];
14     for (k = 0; k < K; k++)
15         q[l][k] /= z;
16 }
```

$ap[k]$ is the second term in (18).

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

2.3.1 Derivation

Again, ELBO is

$$F[q(\mathbf{z}, \boldsymbol{\theta})] = \int \sum_{d=1}^M \sum_{i=1}^{N_d} q(z_{d,i}) q(\boldsymbol{\theta}_d) \log p(w_{d,i}|z_{d,i}, \beta) p(z_{d,i}|\boldsymbol{\theta}_d) d\boldsymbol{\theta}_d - \sum_{d=1}^M \sum_{i=1}^{N_d} \sum_{k=1}^K q(z_{d,i} = k) \log q(z_{d,i} = k) \quad (19)$$

$$- \underbrace{\sum_{d=1}^M \int q(\boldsymbol{\theta}_d) \log \frac{q(\boldsymbol{\theta}_d)}{p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})} d\boldsymbol{\theta}_d}_{\text{KL}[q(\boldsymbol{\theta}_d)||p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})]}$$

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

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

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

Using variational inference,

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

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

- Dirichlet distribution

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

- 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

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

- Remember $z_{d,i} \sim \text{Multi}(\boldsymbol{\theta}_d)$ (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.

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

Now, we can back to the variational inference

$$q(\boldsymbol{\theta}_d) \propto p(\boldsymbol{\theta}_d | \boldsymbol{\alpha}) \exp \left[\sum_{\mathbf{z}} q(\mathbf{z}) \sum_{i=1}^{N_d} \log p(z_{d,i} | \boldsymbol{\theta}_d) \right] \quad (26)$$

$$\propto \prod_{k=1}^K \theta_{d,k}^{\alpha_k - 1} \exp \left[\sum_{\mathbf{z}} q(\mathbf{z}) \sum_{i=1}^{N_d} \sum_{k=1}^K \delta(z_{d,i} = k) \log \theta_{d,k} \right] \quad (27)$$

$$= \exp \left[\sum_{k=1}^K (\alpha_k - 1) \log \theta_{d,k} \right] \exp \left[\sum_{k=1}^K \sum_{i=1}^{N_d} q(z_{d,i} = k) \log \theta_{d,k} \right] \quad (28)$$

$$= \exp \left[\sum_{k=1}^K (\alpha_k - 1) \log \theta_{d,k} \right] \exp \left[\sum_{k=1}^K \mathbb{E}_{q(\mathbf{z}_d)}[N_{d,k}] \log \theta_{d,k} \right] \quad (29)$$

$$= \exp \left[\sum_{k=1}^K (\mathbb{E}_{q(\mathbf{z}_d)}[N_{d,k}] + \alpha_k - 1) \log \theta_{d,k} \right] \quad (30)$$

$$= \prod_{k=1}^K \theta_{d,k}^{\mathbb{E}_{q(\mathbf{z}_d)}[N_{d,k}] + \alpha_k - 1} \quad (31)$$

From (27) to (28), we marginalize the equation with respect to $q(\mathbf{z})$. For various $z_{d,i}$ in \mathbf{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 $\xi_{d,k}^\theta = \mathbb{E}_{q(\mathbf{z}_d)}[N_{d,k}] + \alpha_k$, $q(\boldsymbol{\theta}_d)$ is a Dirichlet distribution whose parameters are $\boldsymbol{\xi}_d^\theta = (\xi_{d,1}^\theta, \xi_{d,2}^\theta, \dots, \xi_{d,K}^\theta)$. We can easily normalize it:

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

2.3.2 Code

2.3.2.1 $\mathbb{E}_{q(\mathbf{z}_d)}[N_{d,k}]$ 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,

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

In C,

```
1 /* In vbem.c */
2 for (k = 0; k < K; k++) {
3     z = 0;
4     for (l = 0; l < L; l++)
5         z += q[l][k] * d->cnt[l];
6     nt[k] = z;
7 }
```

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

3 More notes on code

3.1 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 (more details are in `LDA-explanation.ipynb`),

```
1 def accum_beta(betas, q, t):
2     # t = d[i]
3     betas[t[0],:] += matrix(diag(t[1])) * q
4     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,

```
1 /* in learn.c */
2 int i, k;
3 int n = dp->len;
4
5 for (i = 0; i < n; i++)
6     for (k = 0; k < K; k++)
7         betas[dp->id[i]][k] += q[i][k] * dp->cnt[i];
```

Original article says

$$\beta_{ij} \propto \sum_{d=1}^M \sum_{n=1}^{N_d} \phi_{dni}^* w_{dn}^j \quad (33)$$

for this part. ϕ_{dni} in the original article (n is the n th 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 i th 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).

```
1 for i in range(n): # n is the number of documents
2     gamma, q = lda.vbem(d[i], beta, alpha, demmax)
3     gammas[i,:] = gamma
4     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

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

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