
Text Mining – Assignment #2

Roger Cuscó, Matthew-Sudmann-Day and Miquel Torrens

Exercise 1

Answer.

Exercise 2

Part (a)

The components of the model are the following:

- Parameters:

- $\boldsymbol{\rho} = (\rho_1, \dots, \rho_k, \dots, \rho_K) \in \mathbb{R}^K$
- $\mathbf{B}^{(1)} = (\beta_{z_1}^{(1)}, \dots, \beta_{z_i}^{(1)}, \dots, \beta_{z_N}^{(1)})$, where $\beta_{z_i}^{(1)} \in \Delta^{V_1-1}, \forall i$
- $\mathbf{B}^{(2)} = (\beta_{z_1}^{(2)}, \dots, \beta_{z_i}^{(2)}, \dots, \beta_{z_N}^{(2)})$, where $\beta_{z_i}^{(2)} \in \Delta^{V_2-1}, \forall i$

For notational purposes, let $\mathbf{B} = \{\mathbf{B}^{(1)}, \mathbf{B}^{(2)}\}$.

- Latent variables:

- $\mathbf{z} = (z_1, \dots, z_i, \dots, z_N) \in \mathbb{R}^N$, for $i \in \{1, \dots, N\}$ observations

- Observed data:

- $\mathbf{x}_i^{(1)} = (x_{i1}^{(1)}, \dots, x_{iv_1}^{(1)}, \dots, x_{iV_1}^{(1)}) \in \mathbb{R}^{V_1}, \forall i \in \{1, \dots, N\}$
- $\mathbf{x}_i^{(2)} = (x_{i1}^{(2)}, \dots, x_{iv_2}^{(2)}, \dots, x_{iV_2}^{(2)}) \in \mathbb{R}^{V_2}, \forall i \in \{1, \dots, N\}$

For notational purposes, let $\mathbf{x}_i = \{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\} \forall i$ and $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N\}$.

Part (b)

The density function is:

$$\begin{aligned}
 \mathbb{P}(\mathbf{x}_i | \boldsymbol{\rho}, \mathbf{B}) &= \mathbb{P}(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)} | \boldsymbol{\rho}, \mathbf{B}) \\
 &= \mathbb{P}(\mathbf{x}_i^{(1)} | \boldsymbol{\rho}, \mathbf{B}) \mathbb{P}(\mathbf{x}_i^{(2)} | \boldsymbol{\rho}, \mathbf{B}) \quad (\text{By independence of the two vectors}) \\
 &= \sum_{k=1}^K \mathbb{P}(z_k | \boldsymbol{\rho}, \mathbf{B}) \mathbb{P}(\mathbf{x}_i^{(1)} | z_k, \boldsymbol{\rho}, \mathbf{B}) \mathbb{P}(\mathbf{x}_i^{(2)} | z_k, \boldsymbol{\rho}, \mathbf{B}) \\
 &= \sum_{k=1}^K \rho_k \mathbb{P}(\mathbf{x}_i^{(1)} | z_k, \boldsymbol{\rho}, \mathbf{B}) \mathbb{P}(\mathbf{x}_i^{(2)} | z_k, \boldsymbol{\rho}, \mathbf{B}) \\
 &= \sum_{k=1}^K \rho_k \prod_{v_1=1}^{V_1} (\beta_{z_i v_1}^{(1)})^{x_{iv_1}^{(1)}} \prod_{v_2=1}^{V_2} (\beta_{z_i v_2}^{(2)})^{x_{iv_2}^{(2)}}.
 \end{aligned}$$

The likelihood function is the joint probability of all \mathbf{x}_i 's:

$$\ell(\mathbf{x} | \boldsymbol{\rho}, \mathbf{B}) = \prod_{i=1}^N \sum_{k=1}^K \rho_k \prod_{v_1=1}^{V_1} (\beta_{z_i v_1}^{(1)})^{x_{iv_1}^{(1)}} \prod_{v_2=1}^{V_2} (\beta_{z_i v_2}^{(2)})^{x_{iv_2}^{(2)}}.$$

The complete likelihood incorporates the latent variable and writes as follows:

$$\ell_{\text{comp}}(\mathbf{x}, \mathbf{z} | \boldsymbol{\rho}, \mathbf{B}) = \prod_{i=1}^N \prod_{k=1}^K \left[\rho_k \prod_{v_1=1}^{V_1} (\beta_{z_i v_1}^{(1)})^{x_{iv_1}^{(1)}} \prod_{v_2=1}^{V_2} (\beta_{z_i v_2}^{(2)})^{x_{iv_2}^{(2)}} \right]^{\mathbb{1}_{\{z_i=k\}}}.$$

Finally, the complete log-likelihood gives:

$$\log \ell_{\text{comp}}(\mathbf{x}, \mathbf{z} | \boldsymbol{\rho}, \mathbf{B}) = \sum_{i=1}^N \sum_{k=1}^K \mathbb{1}_{\{z_i=k\}} \left[\log \rho_k + \sum_{v_1=1}^{V_1} x_{iv_1}^{(1)} \log \beta_{z_i v_1}^{(1)} + \sum_{v_2=1}^{V_2} x_{iv_2}^{(2)} \log \beta_{z_i v_2}^{(2)} \right].$$

Part (c)

Denote the function Q as:

$$Q(\boldsymbol{\rho}, \mathbf{B}) = \sum_{i=1}^N \sum_{k=1}^K \hat{z}_{i,k} \left[\log \rho_k + \sum_{v_1=1}^{V_1} x_{iv_1}^{(1)} \log \beta_{z_i v_1}^{(1)} + \sum_{v_2=1}^{V_2} x_{iv_2}^{(2)} \log \beta_{z_i v_2}^{(2)} \right], \quad (1)$$

where using the Bayes' rule:

$$\begin{aligned} \hat{z}_{i,k} &\equiv \mathbb{E} [\mathbb{1}_{\{z_i=k\}} | \boldsymbol{\rho}, \mathbf{B}, \mathbf{x}_i] = \mathbb{P}(z_i = k | \boldsymbol{\rho}, \mathbf{B}, \mathbf{x}_i) \\ &\propto \mathbb{P}(\mathbf{x}_i | \boldsymbol{\rho}, \mathbf{B}, z_i = k) \mathbb{P}(z_i = k | \boldsymbol{\rho}, \mathbf{B}) \\ &= \rho_k \prod_{v_1=1}^{V_1} \left(\beta_{z_i v_1}^{(1)} \right)^{x_{iv_1}^{(1)}} \prod_{v_2=1}^{V_2} \left(\beta_{z_i v_2}^{(2)} \right)^{x_{iv_2}^{(2)}}. \end{aligned} \quad (2)$$

Part (d)

The Lagrangian is defined as:

$$\mathcal{L}(\boldsymbol{\rho}, \mathbf{B}, \boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \nu) := Q(\boldsymbol{\rho}, \mathbf{B}) + \nu \left(1 - \sum_{k=1}^K \rho_k \right) + \sum_{k=1}^K \lambda_k^{(1)} \left(1 - \sum_{v_1=1}^{V_1} \beta_{z_i v_1}^{(1)} \right) + \sum_{k=1}^K \lambda_k^{(2)} \left(1 - \sum_{v_2=1}^{V_2} \beta_{z_i v_2}^{(2)} \right).$$

We take derivatives with respect to the parameters:

$$\frac{\partial \mathcal{L}(\cdot)}{\partial \rho_k} = 0 \Leftrightarrow \sum_{i=1}^N \frac{\hat{z}_{i,k}}{\rho_k} - \nu = 0 \quad (3)$$

$$\frac{\partial \mathcal{L}(\cdot)}{\partial \beta_{k,v_1}} = 0 \Leftrightarrow \sum_{i=1}^N \hat{z}_{i,k} x_{iv_1}^{(1)} \frac{1}{\beta_{z_i v_1}^{(1)}} - \lambda_k^{(1)} = 0 \quad (4)$$

$$\frac{\partial \mathcal{L}(\cdot)}{\partial \beta_{k,v_2}} = 0 \Leftrightarrow \sum_{i=1}^N \hat{z}_{i,k} x_{iv_2}^{(2)} \frac{1}{\beta_{z_i v_2}^{(2)}} - \lambda_k^{(2)} = 0 \quad (5)$$

$$\frac{\partial \mathcal{L}(\cdot)}{\partial \nu} = 0 \Leftrightarrow \sum_{k=1}^K \rho_k = 1 \quad (6)$$

$$\frac{\partial \mathcal{L}(\cdot)}{\partial \lambda_k^{(1)}} = 0 \Leftrightarrow \sum_{v_1=1}^{V_1} \beta_{z_i v_1}^{(1)} = 1 \quad (7)$$

$$\frac{\partial \mathcal{L}(\cdot)}{\partial \lambda_k^{(2)}} = 0 \Leftrightarrow \sum_{v_2=1}^{V_2} \beta_{z_i v_2}^{(2)} = 1. \quad (8)$$

Combining (3) and (6) we see:

$$\nu = \frac{1}{\rho_k} \sum_{i=1}^N \hat{z}_{i,k} \Leftrightarrow \underbrace{\nu \sum_{k=1}^K \rho_k}_{=1} = \sum_{k=1}^K \sum_{i=1}^N \hat{z}_{i,k} \underbrace{\Leftrightarrow}_{\text{Plug } \nu \text{ into (3)}} \rho_k^* = \frac{\sum_{i=1}^N \hat{z}_{i,k}}{\sum_{k=1}^K \sum_{i=1}^N \hat{z}_{i,k}}, \quad (9)$$

Analogously, combining (4) and (7) we get:

$$\beta_{z_i v_1}^{(1)} \lambda_k^{(1)} = \sum_{i=1}^N \hat{z}_{i,k} x_{iv_1}^{(1)} \Leftrightarrow \underbrace{\sum_{v_1=1}^{V_1} \beta_{z_i v_1}^{(1)} \lambda_k^{(1)}}_{=1} = \sum_{i=1}^N \hat{z}_{i,k} \sum_{v_1=1}^{V_1} x_{iv_1}^{(1)} \xLeftrightarrow{\text{Plug } \lambda_k^{(1)} \text{ into (4)}} \left(\beta_{z_i v_1}^{(1)} \right)^* = \frac{\sum_{i=1}^N \hat{z}_{i,k} x_{iv_1}^{(1)}}{\sum_{i=1}^N \hat{z}_{i,k} \sum_{v_1=1}^{V_1} x_{iv_1}^{(1)}}. \quad (10)$$

The exact same procedure can be applied with (5) and (8) (simply changing subscripts) to obtain:

$$\beta_{z_i v_1}^{(2)} \lambda_k^{(2)} = \sum_{i=1}^N \hat{z}_{i,k} x_{iv_2}^{(2)} \Leftrightarrow \underbrace{\sum_{v_2=1}^{V_2} \beta_{z_i v_2}^{(2)} \lambda_k^{(2)}}_{=1} = \sum_{i=1}^N \hat{z}_{i,k} \sum_{v_2=1}^{V_2} x_{iv_2}^{(2)} \xLeftrightarrow{\text{Plug } \lambda_k^{(2)} \text{ into (5)}} \left(\beta_{z_i v_2}^{(2)} \right)^* = \frac{\sum_{i=1}^N \hat{z}_{i,k} x_{iv_2}^{(2)}}{\sum_{i=1}^N \hat{z}_{i,k} \sum_{v_2=1}^{V_2} x_{iv_2}^{(2)}}. \quad (11)$$

The parameters found in this step, i.e. the set of ρ_k^* 's, $\left(\beta_{z_i v_1}^{(1)} \right)^*$'s and $\left(\beta_{z_i v_2}^{(2)} \right)^*$'s will be the ones used in the next iteration.

Part (e)

Pseudo-code for the EM algorithm in this context:

```
#####
em = function(X, K, eps = 1e-6, maxiter = 1e5) {
# X: input data, with word counts for the N observations
# K: number of clusters of the latent variable
# eps: value that defines the distance at which the algorithm stabilizes
# maxiter: maximum number of iterations allowed before convergence
#####
# Number of observations
N = length(D)

# Initialize parameters
for (i in 1:N) {
  beta1[i] = [1 / v1, ..., 1 / V1] # length V1
  beta2[i] = [1 / v2, ..., 1 / V2] # length V2
}
B1 = [beta1[1], ..., beta1[N]] # matrix with betas1
B2 = [beta2[1], ..., beta2[N]] # matrix with betas2
rho = [1 / K, ..., 1 / K] # length K

# Perform iteration until convergence
iter = 1
repeat {
  # Expectation step
  zhat = compute[Equation (2); X, rho, B1, B2]
  Q = compute[Equation (1); X, rho, B1, B2]

  # Maximization step
  for (k in 1:K) {
    rho[k] = sum(zhat[1:N, k]) / sum(zhat[1:N, 1:K]) # Equation (9)
  }
}
```

```

for (i in 1:N) {
  # Equation (10)
  for (v1 in 1:V1) {
    beta1[i, v1] = sum(zhat[1:N, k] * X[1:N, v1]) /
                  sum(zhat[1:N, k] * X[1:N, 1:V1])
  }

  # Equation (11)
  for (v2 in 1:V2) {
    beta1[i, v2] = sum(zhat[1:N, k] * X[1:N, v2]) /
                  sum(zhat[1:N, k] * X[1:N, 1:V2])
  }
}

# Update all parameters
B1 = [beta1[1], ..., beta1[N]] # matrix with betas1
B2 = [beta2[1], ..., beta2[N]] # matrix with betas2
rho = [1 / K, ..., 1 / K] # length K

# If the parameters have stabilized, we stop
if (norm(B1) < eps & norm(B2) < eps & norm(rho) < eps) {
  converged = TRUE
  break
}

# If the algorithm has not stabilized after lots of iterations, stop
if (iter > maxiter) {
  converged = FALSE
  break
}

# Update iterator
iter = iter + 1
}

# End
return(zhat, B1, B2, rho, converged)
}

```