5
Latent Variable Models for Document Analysis

Document Clustering

Given a collection of documents $\{d_1, \ldots, d_N\}$ we would like to partition them into K clusters.

Document Representation. Each document d_n is made of some text. We treat a document as a *set* of words in its text irrespective of their positions, i.e. a document consisting of the text "Andi went to school to meet Bob" is represented as the following set of words: {Andi, went, to, school, to, meet, Bob}. Note that if we had another document with the text "Bob went to school to meet Andi", it would have the same representation as the previous sentence. We refer to this as the *bag of word* representation of the document. Also, we assume the words appearing in our collection of documents come from a dictionary denoted by $\mathcal{A}$.

Generative Model. We assume the following hypothetical generative story for generating the collection of our documents:

- For each document d_n
  - toss the K faced dice with the parameter $\boldsymbol{\varphi}$ to choose the face (i.e. the cluster) k that the nth document belongs to
  - For each word placeholder in the document d_n
    - generate the word by tossing the dice with parameter $\boldsymbol{\mu}_k$ corresponding to the face k.

The parameters of the model are (1) the clusters proportion $\boldsymbol{\varphi}$ which is a probability vector of size K $\sum_{k=1}^{K} \varphi_k = 1$ where , and (2) the word proportion $\boldsymbol{\mu}_k$ corresponding to the kth face of the dice where $\sum_{w \in \mathcal{A}} \mu_{k,w} = 1$; note that we have K such word proportion vectors each of which corresponding to a face of the dice (or cluster).

The probability of generating a a pair of a document and its cluster (k,d), according to our generative story, is

$$\begin{aligned} p(k,d) = p(k)p(d|k) &= \varphi_k \prod_{w \in d} \mu_{k,w} \\ &= \varphi_k \prod_{w \in \mathcal{A}} \mu_{k,w}^{c(w,d)} \end{aligned}$$

where $c(w,d)$ is simply the number of occurrences of the word w in the document d.

In practice, the document cluster ids are not given to us, so we use latent variables to denote the cluster assignments.

Complete Data

The Likelihood. Assume that in addition to the documents $\{d_1, \ldots, d_N\}$, we were also given the values of the corresponding latent variables $\{\boldsymbol{z}_1, .., \boldsymbol{z}_N\}$, then

$$p(d_1, \boldsymbol{z}_1, \ldots, d_N, \boldsymbol{z}_N) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\varphi_{k_n} \prod_{w \in \mathcal{A}} \mu_{k_n,w}^{c(w,d)} \right)^{z_{n,k}}$$

where $\boldsymbol{z}_n := (z_{n1}, \ldots, z_{nk})$ is the cluster assignment vector for the nth document in which $z_{nk} = 1$ if this document belongs to the cluster k and zero otherwise. Note that only one element of the cluster assignment vector is 1, and the rest are zero. To summarise, the log-likelihood of the complete data is:

$$\ln p(d_1, \boldsymbol{z}_1, \ldots, d_N, \boldsymbol{z}_N) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{n,k} \left(\ln \varphi_{k_n} + \sum_{w \in \mathcal{A}} c(w,d) \ln \mu_{k_n,w} \right)$$

Learning the Parameters. Maximising the complete data log-likelihood to learn the parameters of the model leads to the following solution:

- The mixing components: $\varphi_k = \dfrac{N_k}{N}$ where $N_k := \sum_{n=1}^{N} z_{nk}$
- The word proportion parameters for each cluster: $\mu_{k,w} = \dfrac{\sum_{n=1}^{N} z_{n,k} c(w, d_n)}{\sum_{w' \in \mathcal{A}} \sum_{n=1}^{N} z_{n,k} c(w', d_n)}$

The above results can be obtained by forming the Lagrangian (to enforce the constraints, e.g. the sum of mixing coefficients must be zero), and then setting the derivatives with respect to the parameters to zero (see the Appendix A).

The above estimates for the optimal parameters are very intuitive. For example, the best value for $\boldsymbol{\mu}_k$ is obtained by counting the number of times that each word of the dictionary has been seen in the documents belonging to the cluster k, and then normalising this count vector so that it sums to 1 (be dividing each element of the count vector by the sum of the counts).

Incomplete Data and EM

The Likelihood. In practice, the document clusters are not given to us, so $\boldsymbol{z}_n$ is latent. So, the probability of the observed documents is

$$p(d_1, \ldots, d_N) = \prod_{n=1}^{N} p(d_n) = \prod_{n=1}^{N} \sum_{k=1}^{K} p(z_{n,k} = 1, d_n) = \prod_{n=1}^{N} \sum_{k=1}^{K} \left(\varphi_k \prod_{w \in \mathcal{A}} \mu_{k,w}^{c(w,d_n)} \right)$$

To summarise, the log-likelihood is:

$$\ln p(d_1, \ldots, d_N) = \sum_{n=1}^{N} \ln p(d_n) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} p(z_{n,k} = 1, d_n) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \left(\varphi_k \prod_{w \in \mathcal{A}} \mu_{k,w}^{c(w,d_n)} \right)$$

To maximise the above incomplete data log-likelihood objective, we resort to the EM Algorithm.

The Q Function. Let's look at the Q function, which will be the basis of our EM Algorithm:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) := \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_{n,k} = 1 | d_n, \boldsymbol{\theta}^{\text{old}}) \ln p(z_{n,k} = 1, d_n | \boldsymbol{\theta})$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_{n,k} = 1 | d_n, \boldsymbol{\theta}^{\text{old}}) \left(\ln \varphi_k + \sum_{w \in \mathcal{A}} c(w, d_n) \ln \mu_{k,w} \right)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{n,k}) \left(\ln \varphi_k + \sum_{w \in \mathcal{A}} c(w, d_n) \ln \mu_{k,w} \right)$$

where $\boldsymbol{\theta} := (\boldsymbol{\varphi}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$ is the collection of model parameters, and $\gamma(z_n, k) := p(z_{n,k} = 1 | d_n, \boldsymbol{\theta}^{\text{old}})$ are the responsibility factors.

To maximise the Q function, we can form the Lagrangian to enforce the constraints (See Appendix A), and set the derivatives to zero which leads to the following solution for the model parameters:

- The mixing components: $\varphi_k = \dfrac{N_k}{N}$ where $N_k := \sum_{n=1}^{N} \gamma(z_{n,k})$
- The word proportion parameters for each cluster: $\mu_{k,w} = \dfrac{\sum_{n=1}^{N} \gamma(z_{n,k}) c(w, d_n)}{\sum_{w' \in \mathcal{A}} \sum_{n=1}^{N} \gamma(z_{n,k}) c(w', d_n)}$

The EM Algorithm. Now let's put everything together to construct our EM algorithm to learn the parameters and find the best value for the latent variables:

- Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}} = (\boldsymbol{\varphi}^{\text{old}}, \boldsymbol{\mu}_1^{\text{old}}, \ldots, \boldsymbol{\mu}_K^{\text{old}})$

- While the convergence is not met:
  - **E step:** Set $\forall n, \forall k : \gamma(z_{n,k})$ based on $\boldsymbol{\theta}^{\text{old}}$
  - **M Step:** Set $\boldsymbol{\theta}^{\text{new}}$ based on $\forall n, \forall k : \gamma(z_{n,k})$
  - $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$