MONASH University
Information Technology

FIT5201

Data Analysis Algorithms

Document Clustering and Introduction to Neural Networks

Document Clustering

- Given a collection of documents $\{d_1, d_2, \dots, d_N\}$ we would like to partition them into K clusters.
- Document representation
  - Each document is made of some text
  - *bag of word* representation of the document
  - We treat a document as a *set* of words in its text irrespective of their positions
  - Also, we assume the words appearing in our collection of documents come from a dictionary denoted by $\mathcal{A}$

Bag of Words

```
(1) John likes to watch movies. Mary likes movies too.
```

```
(2) John also likes to watch football games.
```

```
[
    "John",
    "likes",
    "to",
    "watch",
    "movies",
    "Mary",
    "too",
    "also",
    "football",
    "games"
]
```

```
(1) [1, 2, 1, 1, 2, 1, 1, 0, 0, 0]
(2) [1, 1, 1, 1, 0, 0, 0, 1, 1, 1]
```

Understanding the Model in Alexandria: an example

- d_1=this one has a little star
- d_2=this one has a little car
- d_3=I would not like them here or there
- d_4=I would not like them anywhere
- d_5=I do not like green eggs and ham

- Assume we know the clusters beforehand (in reality we don't)
  - K=2 (two clusters from two books)
  - C_1=(d_1, d_2), C_2=(d_3, d_4, d_5)
  - $\varphi_1 = 0.4$ (2/5), $\varphi_2 = 0.6$ (3/5)
  - Dictionary for C_1=(this, one, has, little, star, car)
  - μ_1=(2/10, 2/10, 2/10, 2/10, 1/10, 1/10) = (0.2, 0.2, 0.2, 0.2, 0.1, 0.1)
  - Dictionary for C_2=(I, would, not, like, them, here, there, anywhere, do, green, eggs, ham)
  - μ_2=(0.15, 0.1, 0.15, 0.15, 0.1, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)

Quotes from Dr Suess's books *One Fish, Two Fish* and *Green Eggs and Ham*

Generating Words

- $\mathcal{A}$ ={this, one, has, little, star, car, I, would, not, like, them, here, there, anywhere, do, green, eggs, ham}
- P(this, one, has, little, star) = P(this)P(one)P(has) P(little) P(star)
- $P(d|k) = \prod_{w \in d} P(w|k) = \prod_{w \in \mathcal{A}} P(w|k)^{c(w,d)}$

Generative Model

- For each document d_n
  - Toss the K-face dice (with the probability parameter φ) to choose the face k (i.e., the cluster) that the n^{th} document belongs to
  - For each word placeholder in the document d_n
    - > Generate the word by tossing the dice (with the probability parameter μ_k) corresponding to the face k

- Parameters:
  - The clusters proportion $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_K), \varphi_k \geq 0, \sum_{k=1}^{K} \varphi_k = 1$
  - The word proportion $\mu_k = \left(\mu_{k,1}, \mu_{k,2}, \dots, \mu_{k,|\mathcal{A}|}\right), \mu_{k,w} \geq 0, \sum_{w \in \mathcal{A}} \mu_{k,w} = 1$
  - These are constraints which allow us to use Lagrange model to learn the parameters

Generative Model

- The probability of generating a document and its cluster (k, d) is

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

- $c(w, d)$ is the number of occurrences of the word w in the document d
- In practice,
  - The document cluster labels are not given to us

Complete Data

- Documents $\{d_1, d_2, \dots, d_N\}$
- We use latent variables $\boldsymbol{z}_n$ to denote the cluster assignments for n^{th} document
- $\boldsymbol{z}_n = (z_{n1}, z_{n2}, \dots, z_{nK})$
  - $z_{nk} = \begin{cases} 1, & d_n \in C_k \\ 0, & d_n \notin C_k \end{cases}$
  - Only one element in z_{nk} is 1. The rest are zero

Complete Data

$$p(d_1, z_1, \dots, d_N, z_N) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\varphi_k \prod_{w \in \mathcal{A}} \mu_{k,w}^{c(w,d_n)} \right)^{z_{nk}}$$

$$z_{nk} = \begin{cases} 1, & d_n \in \mathcal{C}_k \\ 0, & d_n \notin \mathcal{C}_k \end{cases}$$

- With the constraint that $\sum_{k=1}^{K} \varphi_k = 1$ and $\sum_{w \in \mathcal{A}} \mu_{k,w} = 1$
- Use Lagrange model to solve the parameters
  - Constrained optimization: convert it to unconstrained optimization problems which can be solved either find a solution analytically or use an iterative algorithm to find a solution
  - Lagrange model

Complete Data

- Use Lagrange model to solve the parameters
  - Constrained optimization: convert it to unconstrained optimization problems which can be solved either finding a solution analytically or using an iterative algorithm to find a solution
  - Lagrange multipliers

$$\begin{array}{ll} \underset{x}{\text{maximise}} & f(x) \\ \text{subject to} & g_i(x) = 0 \quad i = 1, \ldots, m \end{array}$$

    > Equality constraints

$$\mathcal{L}(x, \lambda_1, \ldots, \lambda_m) := f(x) - \lambda_1 g_1(x) - \ldots - \lambda_m g_m(x)$$

    > The stationary points for $f(x)$ are ensured to be the stationary points for the new function, but not conversely

Complete Data…

- Through the Lagrange multiplier on Maximum Likelihood Function

Mixing components: $\varphi_k = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N} z_{nk}$

Word proportion parameters: $\mu_{kw} = \frac{\sum_{n=1}^{N} z_{nk} c(w, d_n)}{\sum_{w\prime \in \mathcal{A}} \sum_{n=1}^{N} z_{nk} c(w\prime, d_n)}$

Incomplete Data and EM

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

- Hard to derive the analytical solutions
- Resort to EM algorithm

Refreshment about GMMs

- Training objective: find maximum likelihood solution for models having latent variables.
  - Observed data X, Latent variable Z, set of model parameters θ
  - Log likelihood function

$$\ln p(X|\theta) = \ln \sum_{Z} p(X, Z|\theta)$$

Refreshment about GMMs

$$\gamma(z_{nk}) = p(z_n = k|x_n) = \frac{\varphi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \varphi_j N(x_n|\mu_j, \Sigma_j)}$$

- $\gamma(z_{nk})$: posterior probability once we observed x_n
- $\sum_{k=1}^{K} \gamma(z_{nk}) = 1$
- Partial assignment or soft assignment
- φ_k prior probability of $z_n = k$
- We do simultaneously
  - Cluster prediction and parameter estimation
  - Use iterative Expectation Maximisation (EM)

Refreshment about GMMs

- Training objective: find maximum likelihood solution for models having latent variables.
  - Observed data X, Latent variable Z, set of model parameters θ
  - Log likelihood function

$$\ln p(X|\theta) = \ln \sum_{Z} p(X, Z|\theta)$$

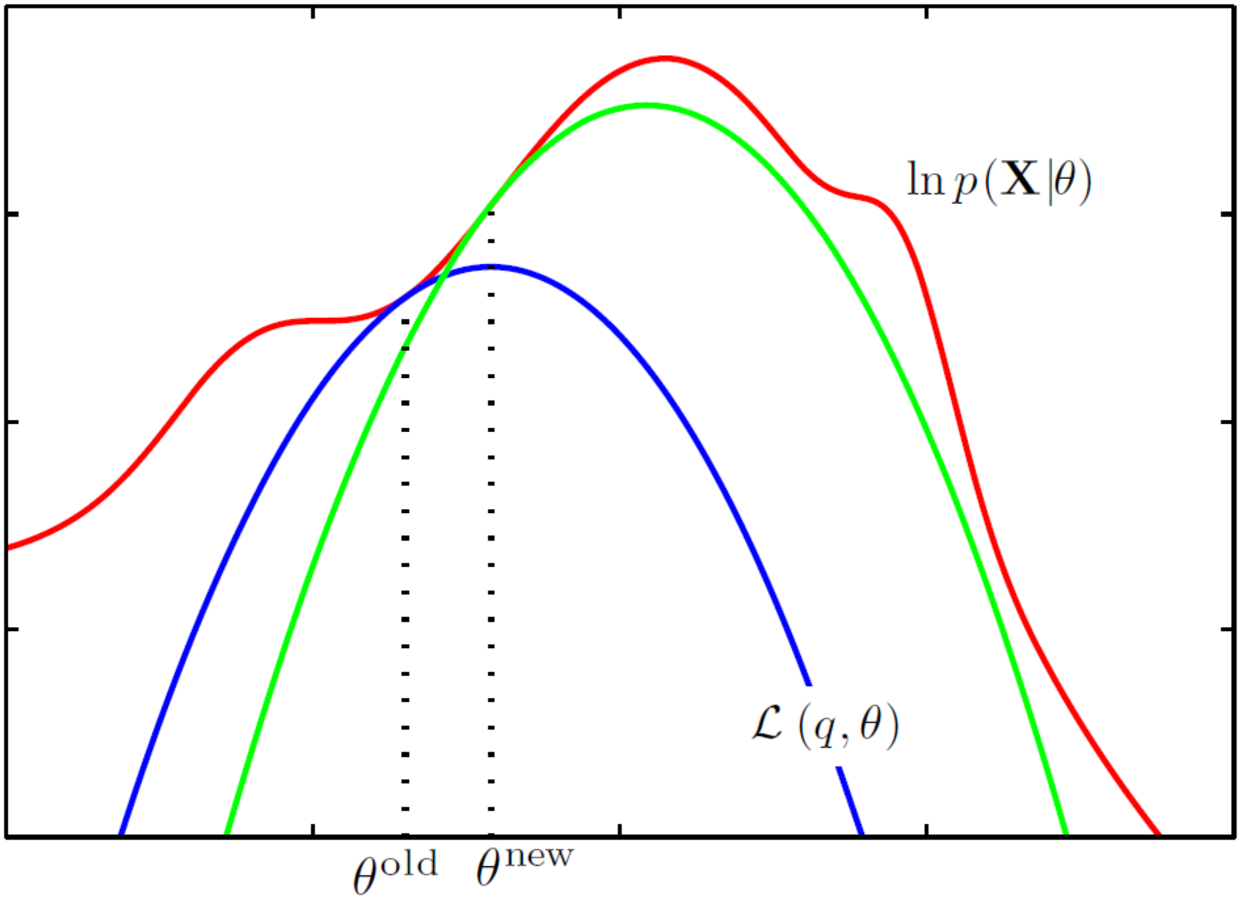
- Algorithm:
  - Choose an initial setting for the parameters θ^{old}
  - While convergence is not met:
    > **E Step**: Evaluate $p\left(Z|X, \theta^{old}\right)$
    > **M Step**: Evaluate θ^{new} given by

$$\theta^{new} \leftarrow \arg\max_{\theta} \underbrace{\sum_{Z} p\left(Z|X, \theta^{old}\right) \ln p(X, Z|\theta)}_{Q\left(\theta, \theta^{old}\right)}$$

    > $\theta^{old} \leftarrow \theta^{new}$

Refreshment about GMMs

- Is each iteration guaranteed to increase the log likelihood function?
- What's the relationship between the Q function and log likelihood function?



Incomplete Data and EM

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

- Q function

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

$$\gamma(z_n, k) := p(z_{n,k} = 1 | d_n, \boldsymbol{\theta}^{\text{old}})$$

Incomplete Data and EM

$$\varphi_k = \frac{N_k}{N} \text{ where } N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\mu_{kw} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})\, c(w, d_n)}{\sum_{w' \in \mathcal{A}} \sum_{n=1}^{N} \gamma(z_{nk})\, c(w', d_n)}$$

- 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 the above equations
  - $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$

Other Methods for Document Clustering

- Other methods can be used to encode the documents, e.g., TF-IDF
- Use Euclidian distance to measure the similarity between documents/cluster vectors
- Can use K-Means to cluster the documents into K clusters
  - What we do in the tutorial