ML Homework 3

Samuel Pegg

December 2020

1 Clustering: Mixture of Multinomials

1.1 MLE for Multinomial

For $i = 1, \ldots, d$

$$P(\boldsymbol{x}, \boldsymbol{\mu}) = \frac{n!}{\prod_j x_j!} \prod_i \mu_i^{x_i}$$

Define $\boldsymbol{\mu}_{ML}$ to be the MLE estimate of $\boldsymbol{\mu}$. Then

$$\begin{aligned}
\boldsymbol{\mu}_{ML} = f(\boldsymbol{\mu}) :&= \arg\max_{\boldsymbol{\mu}} \log \left(\frac{n!}{\prod_j x_j!} \prod_i \mu_i^{x_i} \right) \\
&= \arg\max_{\boldsymbol{\mu}} \left\{ \log(n!) - \sum_j \log(x_j!) + \sum_i x_i \log(\mu_i) \right\} \\
&= \arg\min_{\boldsymbol{\mu}} \left\{ -\log(n!) + \sum_j \log(x_j!) - \sum_i x_i \log(\mu_i) \right\}
\end{aligned}$$

subject to the constraints

$$\sum_i \mu_i = 1, \qquad \mu_i \in (0,1)\ \forall i, \qquad \sum_i x_i = n$$

We formulate the Lagrangian:

$$\mathcal{L}(\boldsymbol{\mu}, \lambda) = -\log(n!) + \sum_j \log(x_j!) - \sum_i x_i \log(\mu_i) + \lambda \left(\sum_i \mu_i - 1 \right)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_k} = \lambda - \frac{x_k}{\mu_k} = 0 \quad \implies \mu_k = \frac{x_k}{\lambda} \quad \implies \boldsymbol{\mu} = \frac{\boldsymbol{x}}{\lambda}$$

Since $\sum_i \mu_i = 1$,

$$\sum_i \frac{x_i}{\lambda} = 1 \quad \implies \lambda = \sum_i x_i = n \quad \implies \boldsymbol{\mu}_{ML} = \frac{\boldsymbol{x}}{n}$$

1.2 EM for mixture of Multinomials

Since

$$p(d) = \frac{n_d!}{\prod_w T_{dw}!} \sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}}$$

and documents are i.i.d., for the whole corpus D we have

$$p(D) = \prod_d p(d) = \prod_d \frac{n_d!}{\prod_w T_{dw}!} \sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}}$$

and the log likelihood is hence

$$\begin{aligned} \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}) &= \sum_d \log \left(\frac{n_d!}{\prod_w T_{dw}!} \sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}} \right) \\ &= \sum_d \left\{ \log \left(\frac{n_d!}{\prod_w T_{dw}!} \right) + \log \left(\sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}} \right) \right\} \end{aligned}$$

Since probabilities sum to 1, we have the constraints

$$\sum_k \pi_k = 1 \quad \text{and} \quad \sum_w \mu_{wk} = 1 \text{ for } k = 1, \dots, K$$

Thus:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \lambda_\pi) = \sum_d \left\{ \log \left(\frac{n_d!}{\prod_w T_{dw}!} \right) + \log \left(\sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}} \right) \right\} \\ + \lambda_\pi \left(\sum_k \pi_k - 1 \right) + \sum_{k=1}^{K} \lambda_k \left(1 - \sum_w \mu_{wk} \right) \end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial \pi_j} = \sum_d \frac{\prod_w \mu_{wj}^{T_{dw}}}{\sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}}} + \lambda_\pi = 0$$

Let

$$\gamma(z_{jd}) = \frac{\pi_j \prod_w \mu_{wj}^{T_{dw}}}{\sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}}} \quad \text{and} \quad N_j = \sum_d \gamma(z_{jd}) \implies \sum_{j=1}^{K} N_j = D$$

Then

$$\sum_d \frac{\gamma(z_{jd})}{\pi_j} = -\lambda_\pi \implies \sum_j \pi_j = 1 = -\frac{1}{\lambda_\pi} \sum_j N_j = -\frac{D}{\lambda_\pi} \implies \lambda_\pi = -D$$

$$\implies \pi_j = \frac{N_j}{D}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mu_{ij}} = \sum_d \frac{T_{di} \mu_{ij}^{T_{di}-1} \pi_j \prod_{w \neq i} \mu_{wj}^{T_{dw}}}{\sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}}} - \lambda_j &= \frac{1}{\mu_{ij}} \sum_d \frac{T_{di} \pi_j \prod_w \mu_{wj}^{T_{dw}}}{\sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}}} - \lambda_j \\ &= \frac{1}{\mu_{ij}} \sum_d T_{di} \gamma(z_{jd}) - \lambda_j = 0 \end{aligned}$$

$$
\begin{aligned}
\implies \sum_d T_{di}\gamma(z_{jd}) = \lambda_j \mu_{ij} \implies & \sum_i \sum_d T_{di}\gamma(z_{jd}) = \lambda_j \sum_i \mu_{ij} \\
\implies & \lambda_j = \sum_i \sum_d T_{di}\gamma(z_{jd}) = \sum_d n_d \gamma(z_{jd}) \\
\implies & \mu_{ij} = \frac{\sum_d T_{di}\gamma(z_{jd})}{\sum_d n_d \gamma(z_{jd})}
\end{aligned}
$$

Hence the EM algorithm for a Mixture of Multinomials is defined as follows:

1. The E Step – use the current values of $\boldsymbol{\pi}$ and $\boldsymbol{\mu}_j$ for $k = 1, \dots, K$ to calculate:

$$
\gamma(z_{jd}) = \frac{\pi_j \prod_w \mu_{wj}^{T_{dw}}}{\sum_{k=1}^K \pi_k \prod_w \mu_{wk}^{T_{dw}}} \text{ for } j = 1, \dots, K \text{ and } d = 1, \dots, D
$$

2. The M Step – use the result of the E step to calculate:

$$
\begin{aligned}
\pi_j &= \frac{N_j}{D} \text{ for } j = 1, \dots, K \\
\mu_{ij} &= \frac{\sum_d T_{di}\gamma(z_{jd})}{\sum_d n_d \gamma(z_{jd})} \text{ for } i = 1, \dots, W \text{ and } j = 1, \dots, K
\end{aligned}
$$

2 PCA Minimum Error Formulation

Let $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_p$ be a complete orthonormal basis of the data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$. Then for all $n \in 1, \dots, N$, $\boldsymbol{x}_n$ can be expressed as the linear combination

$$
\boldsymbol{x}_n = \sum_i \alpha_{ni} \boldsymbol{\mu}_i \implies \boldsymbol{x}_n^T \boldsymbol{\mu}_j = \alpha_{nj} \implies \boldsymbol{x}_n = \sum_i (\boldsymbol{x}_n^T \boldsymbol{\mu}_i)\boldsymbol{\mu}_i
$$

Now consider a d dimensional approximation of x where $d < p$:

$$
\tilde{\boldsymbol{x}}_n = \sum_{i=1}^d z_{ni}\boldsymbol{\mu}_i + \sum_{i=d+1}^p b_i \boldsymbol{\mu}_i
$$

where we work under the assumption that the b_i are shared across all data points. We define mean squared error as J:

$$
J = \frac{1}{N} \sum_{j=1}^N \|\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_j\|^2
$$

Minimising the error corresponds to the optimisation problem

$$
\min_{\boldsymbol{\mu}, \boldsymbol{z}, \boldsymbol{b}} J \quad \text{subject to} \quad \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i = 1 \ \forall i
$$

Firstly, Lets rewrite the contents of the norm.

$$
\begin{aligned}
\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_j &= \sum_i (\boldsymbol{x}_j^T \boldsymbol{\mu}_i)\boldsymbol{\mu}_i - \sum_{i=1}^d z_{ji}\boldsymbol{\mu}_i - \sum_{i=d+1}^p b_i \boldsymbol{\mu}_i \\
&= \sum_{i=1}^d (\boldsymbol{x}_j^T \boldsymbol{\mu}_i - z_{ji})\boldsymbol{\mu}_i + \sum_{i=d+1}^p (\boldsymbol{x}_j^T \boldsymbol{\mu}_i - b_i)\boldsymbol{\mu}_i
\end{aligned}
$$

$$\implies \qquad \|\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_j\|^2 = \sum_{i=1}^{d} (\boldsymbol{x}_j^T \boldsymbol{\mu}_i - z_{ji})^2 + \sum_{i=d+1} (\boldsymbol{x}_j^T \boldsymbol{\mu}_i - b_i)^2$$

$$\implies \qquad J = \frac{1}{N} \sum_{j=1}^{N} \left\{ \sum_{i=1}^{d} (\boldsymbol{x}_j^T \boldsymbol{\mu}_i - z_{ji})^2 + \sum_{i=d+1} (\boldsymbol{x}_j^T \boldsymbol{\mu}_i - b_i)^2 \right\}$$

Differentiating with respect to z_{ij} and b_i, we get

$$\frac{\partial J}{\partial z_{ji}} = -\frac{2}{N} (\boldsymbol{x}_j^T \boldsymbol{\mu}_i - z_{ji}) = 0 \qquad \implies \qquad z_{ij} = \boldsymbol{x}_j^T \boldsymbol{\mu}_i$$

And

$$\frac{\partial J}{\partial b_i} = -\frac{2}{N} \sum_{j=1}^{N} (\boldsymbol{x}_j^T \boldsymbol{\mu}_i - b_i) = 0$$

$$\implies \qquad 2b_i = \frac{2}{N} \sum_{j=1}^{N} \boldsymbol{x}_j^T \boldsymbol{\mu}_i = \frac{2\boldsymbol{\mu}_i^T}{N} \sum_{j=1}^{N} \boldsymbol{x}_j = 2\boldsymbol{\mu}_i^T \overline{\boldsymbol{x}} \qquad \implies \qquad b_i = \overline{\boldsymbol{x}}^T \boldsymbol{\mu}_i$$

So we can simplify the expression for $\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_j$ to

$$\begin{aligned} \boldsymbol{x}_j - \tilde{\boldsymbol{x}}_j &= \sum_{i=d+1}^{p} \big((\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T \boldsymbol{\mu}_i \big) \boldsymbol{\mu}_i \\ \implies \qquad J &= \frac{1}{N} \sum_j \sum_{i=d+1}^{p} \Big(\big((\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T \boldsymbol{\mu}_i \big) \boldsymbol{\mu}_i \Big)^T \big((\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T \boldsymbol{\mu}_i \big) \boldsymbol{\mu}_i \\ &= \frac{1}{N} \sum_j \sum_{i=d+1}^{p} \big((\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T \boldsymbol{\mu}_i \big)^T \big((\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T \boldsymbol{\mu}_i \big) \\ &= \frac{1}{N} \sum_j \sum_{i=d+1}^{p} \big(\boldsymbol{x}_j^T \boldsymbol{\mu}_i - \overline{\boldsymbol{x}}^T \boldsymbol{\mu}_i \big)^2 \\ &= \sum_{i=d+1}^{p} \boldsymbol{\mu}_i^T S \boldsymbol{\mu}_i \end{aligned}$$

3 Deep Generative Models: Class-conditioned VAE

3.1 Class-conditioned VAE Derivation

The variational lower bound for VAE is

$$L(x) = E\big[\log(p(x|z))\big] - KL\big(q(z|x)||p(z)\big)$$

In the class-conditioned variant, the encoder $q(z|x;\phi)$ becomes $q(z|x,y;\phi)$ and the decoder $p(x|z;\theta)$ becomes $p(x|z,y;\theta)$. Thus the expression for the lower bound becomes

$$L(x,y) = E\big[\log(p(x|z,y))\big] - KL\big(q(z|x,y)||p(z|y)\big)$$

i.e. all of the distributions are now conditioned on y. The real latent variable is distributed under $p(z|y)$, so for each possible value of y we have a unique $p(z)$.

3.2 Implementation in ZhuSuan and Generated Images

To incorporate the new variable y into the VAE code, we can just concatenate. After 30 epochs, the lower bound was -92.06169891357422.