Department of Computer Science and Technology

Machine Learning

Homework 3

Sahand Sabour

2020280401

1 Clustering: Mixture of Multinomials

1.1 MLE for multinomial

The likelihood function for this multinomial distribution is given as

$$P(x|\mu) = \frac{n!}{\prod_i x_i!} \prod_i \mu_i^{x_i}, \quad i = 1, ..., d \tag{1}$$

Taking log from both side of the above equation gives the log-likelihood function

$$\mathcal{L}(\mu) = log(P(x|\mu)) = log(n!) - log(\prod_i x_i!) + log(\prod_i \mu_i^{x_i}) \tag{2}$$

This can be considered a Lagrange problem with the constraint $\sum_i \mu_i = 1$. Hence, the Lagrangian equation can be formulated as

$$\mathcal{L}(\mu) = log(n!) - log(\prod_i x_i!) + log(\prod_i \mu_i^{x_i}) - \lambda(\sum_i \mu_i - 1) \tag{3}$$

where λ is Lagrangian multiplier, giving

$$\mathcal{L}(\mu) = log(n!) - \sum_i log(x_i!) + \sum_i x_i log(\mu_i) - \lambda(\sum_i \mu_i - 1) \tag{4}$$

Taking the derivative of the equation with respect to μ_i and setting it to 0 gives

$$\frac{\partial \mathcal{L}}{\partial \mu_i} = \frac{\sum_i x_i}{\sum_i \mu_i} - \lambda = 0 \tag{5}$$

Hence, we get that

$$\lambda = \frac{\sum_i x_i}{\sum_i \mu_i} = \frac{n}{1} = n \tag{6}$$

Accordingly, we could derive the maximum-likelihood estimator μ_i as

$$\mu_i = \frac{x_i}{\lambda} = \frac{x_i}{n}, \quad i = 1, ..., d \tag{7}$$

1.2 EM for mixture of multinomials

2 PCA

2.1 Minimum Error Formulation

Assuming that we have a set of complete orthonormal basis $\{\mu_i\}$, where $i \in [1, p]$, we have that $\mu_i^T \mu_j = \delta_{ij}$ and each data point can be represented as $x_n = \sum_i a_{ni}\mu_i$. Accordingly, due to orthonormal property, we can get that

$$a_{ni} = x_n^T \mu_i \tag{8}$$

Inserting this in the data point representation gives

$$x_n = \sum_i (x_n^T \mu_i)\mu_i \tag{9}$$

For this approach, the aim is to formulate PCA as minimizing the mean-squared-error of a low-dimensional approximation of the given basis. Hence, we assume a low-dimensional approximation of the point representation as follows

$$\widetilde{x}_n = \sum_i^d z_{ni} + \sum_{i=d+1}^p b_i\mu_i \quad \text{where } b_i \text{ are constants for all points} \tag{10}$$

Therefore, the best approximation is to minimize the following error

$$\min_{U,z,b} J := \frac{1}{N}\sum_{n=1}^N ||x_n - \widetilde{x}_n||^2 \tag{11}$$

Consequently, we have that

$$\begin{aligned} J &= \frac{1}{N}\sum_{n=1}^N ||x_n - \widetilde{x}_n||^2 \\ &= \frac{1}{N}\sum_{n=1}^N (x_n - \widetilde{x}_n)^T(x_n - \widetilde{x}_n) \\ &= \frac{1}{N}\sum_{n=1}^N x_n^T x_n - 2x_n^T\widetilde{x}_n + \widetilde{x}_n^T\widetilde{x}_n \end{aligned}$$

Inserting equation 10 in the above equation and replacing $\widetilde{x}_n$ gives

$$\begin{aligned} J = \frac{1}{N}\sum_{n=1}^N x_n^T x_n - 2x_n^T(\sum_i^d z_{ni}\mu_i + \sum_{i=d+1}^p b_i\mu_i) \\ +(\sum_i^d z_{ni}\mu_i^T + \sum_{i=d+1}^p b_i\mu_i^T)(\sum_i^d z_{ni}\mu_i + \sum_{i=d+1}^p b_i\mu_i) \end{aligned}$$

Accordingly, for minimizing this error, we calculate the derivative with respect to z and b and set it to 0.

$$\frac{\delta J}{\delta z_{nj}} = \frac{1}{n}[-2x_n^T\mu_j + \mu_j^T(\sum_i^d z_{ni}\mu_i + \sum_{i=d+1}^p b_i\mu_i) + (\sum_i^d z_{ni}\mu_i^T + \sum_{i=d+1}^p b_i\mu_i^T)\mu_j] = 0$$

$$\frac{\delta J}{\delta z_{nj}} = \frac{1}{n}[-2x_n^T\mu_j + 2\mu_j^T(\sum_i^d z_{ni}\mu_i + \sum_{i=d+1}^p b_i\mu_i)] = 0$$

$$2\mu_j^T(\sum_i^d z_{ni}\mu_i + \sum_{i=d+1}^p b_i\mu_i) = 2x_n^T\mu_j$$

$$\sum_i^d z_{ni}\mu_j^T\mu_i + \sum_{i=d+1}^p b_i\mu_j^T\mu_i = x_n^T\mu_j$$

$$\sum_i^d z_{ni}\delta ij + \sum_{i=d+1}^p b_i\delta ij = z_{ni} + 0 = x_n^T\mu_i$$

Giving $z_{ni} = x_n^T\mu_i$ for $i \in [1, d]$. Similarly, we the derivative with respect to b

$$\frac{\delta J}{\delta b_j} = \frac{1}{n}\sum[-2x_n^T\mu_j + \mu_j^T(\sum_i^d z_{ni}\mu_i + \sum_{i=d+1}^p b_j\mu_i) + (\sum_i^d z_{ni}\mu_i^T + \sum_{i=d+1}^p b_j\mu_i^T)\mu_j] = 0$$

$$\frac{\delta J}{\delta b_j} = \frac{1}{n}\sum[-2x_n^T\mu_j + 2\mu_j^T(\sum_i^d z_{ni}\mu_i + \sum_{i=d+1}^p b_j\mu_i)] = 0$$

$$\sum(\sum_i^d z_{ni}\mu_j^T\mu_i + \sum_{i=d+1}^p b_j\mu_j^T\mu_i) = \sum x_n^T\mu_j$$

$$\sum b_j = Nb_j = \sum x_n^T\mu_j \quad \text{giving} \quad b_j = \sum \frac{1}{n}x_n^T\mu_j = \bar{x}^T u_j$$

Which in turn gives $b_i = \bar{x}^T u_i$ for $i \in [d+1, p]$. Accordingly, from equation 9, we can get the displacement lines in the orthogonal subspace as follows

$$x_n - \widetilde{x}_n = \sum_{i=d+1}^p \{(x_n - \bar{x})^T\mu_i\}\mu_i \tag{12}$$

Which produces the following optimization problem for error J

$$\min_{\mu_j} J \quad \text{where} \quad \mu_i^T\mu_i = 1 \tag{13}$$

Assuming d=1 (1-dimensional subspace) and p=2 (2-dimensional space), the optimization problem becomes

$$\min_{\mu_2} J = \mu_2^T S \mu_2 \quad \text{where} \quad \mu_2^T \mu_2 = 1 \tag{14}$$

Which gives $S\mu_2 = \lambda_2 \mu_2$, meaning that μ_2 should be chosen as the eigenvector that corresponds to the smaller eigenvalue. Accordingly, the principal subspace is chosen by the eigenvector of the larger eigenvalue.

3 Deep Generative Models: Class-conditioned VAE