A4 Q1

2022年7月12日 星期二　　下午11:58

Ex 1

a)

$$p_\theta(x_i, z_i = k) = \pi_k |S_k|^{-\frac{1}{2}} e^{-\frac{1}{2}(x_i-\mu_k)^T S_k^{-1}(x_i-\mu_k)}$$

$$r_{ik} = q_i(z_i = k)$$

$$\sum_{i=1}^n \sum_{k=1}^k q_i(z_i=k) \log p_\theta(x_i, z_i=k)$$

$$= \sum_{i=1}^n \sum_{k=1}^k r_{ik} \left(\log \pi_k + \log |S_k|^{-\frac{1}{2}} + \log \exp\left(-\frac{1}{2}(x_i-\mu_k)^T S_k^{-1}(x_i-\mu_k)\right)\right)$$

$$= \sum_{i=1}^n \sum_{k=1}^k r_{ik} \left(\log \pi_k - \frac{1}{2} \log \cdot \prod_{j=1}^d S_{kj} - \frac{1}{2}(x_i-\mu_k)^T S_k^{-1}(x_i-\mu_k)\right)$$

$$S_k^{-1} = \begin{bmatrix} \frac{1}{S_{k1}} & & \phi \\ & \frac{1}{S_{k2}} & \\ & & \ddots & \\ \phi & & & \frac{1}{S_{kd}} \end{bmatrix} \quad x_i - \mu_k = \begin{bmatrix} x_{i1} - \mu_{k1} \\ \vdots \\ x_{ij} - \mu_{kj} \\ \vdots \end{bmatrix} \Rightarrow S_k^{-1}(x_i-\mu_k) = \begin{bmatrix} \frac{1}{S_{k1}}(x_{i1}-\mu_{k1}) \\ \vdots \end{bmatrix}$$

$$(x_i-\mu_k)^T S_k^{-1}(x_i-\mu_k) = \begin{bmatrix} x_{i1} - \mu_{k1} \\ \vdots \\ x_{ij} - \mu_{kj} \\ \vdots \end{bmatrix}^T \cdot \begin{bmatrix} \frac{1}{S_{k1}}(x_{i1}-\mu_{k1}) \\ \vdots \end{bmatrix} = \sum_{j=1}^d \frac{1}{S_{kj}} \cdot (x_{ij}-\mu_{kj})^2$$

$$= \sum_{i=1}^n \sum_{k=1}^k r_{ik} \left(\log \pi_k - \frac{1}{2} \sum_{j=1}^d \log S_{kj} - \frac{1}{2} \sum_{j=1}^d \frac{1}{S_{kj}} (x_{ij}-\mu_{kj})^2 \right) = \ell$$

$$\frac{\partial \ell}{\partial \mu_{kj}} = \sum_{i=1}^n r_{ik} \cdot \left(-\frac{1}{S_{kj}}(x_{ij}-\mu_{kj})\right) = 0$$

$$\sum_{i=1}^n r_{ik} x_{ij} = \mu_{kj} \sum_{i=1}^n r_{ik}$$

$$\mu_{kj} = \frac{\sum_{i=1}^n r_{ik} x_{ij}}{\sum_{i=1}^n r_{ik}}$$

$$\frac{\partial \ell}{\partial S_{kj}} = \sum_{i=1}^n r_{ik} \left(-\frac{1}{2} \cdot \frac{1}{S_{kj}} + \frac{1}{2} \frac{1}{S_{kj}^2} (x_{ij}-\mu_{kj})^2 \right) = 0$$

$$\sum_{i=1}^n -r_{ik} S_{kj} + r_{ik}(x_{ij}^2 - 2x_{ij}\mu_{kj} + \mu_{kj}^2) = 0$$

$$\sum_{i=1}^n r_{ik} x_{ij}^2 - 2\mu_{kj} \sum_{i=1}^n r_{ik} x_{ij} + \mu_{kj}^2 \sum_{i=1}^n r_{ik} = S_{kj} \sum_{i=1}^n r_{ik}$$

$$\sum_{i=1} r_{ik} x_{ij}^2 - 2\mu_{kj} \sum_{i=1} r_{ik} x_{ij} + \mu_{kj}^2 \sum_{i=1} r_{ik} = S_{kj} \sum_{i=1} r_{ik}$$

$$S_{kj} = \frac{\sum_{i=1}^n r_{ik} x_{ij}^2}{\sum_{i=1}^n r_{ik}} - 2\mu_{kj} \frac{\sum_{i=1}^n r_{ik} x_{ij}}{\sum_{i=1}^n r_{ik}} + \mu_{kj}^2$$

$$= \frac{\sum_{i=1}^n r_{ik} x_{ij}^2}{\sum_{i=1}^n r_{ik}} - 2\mu_{kj}^2 + \mu_{kj}^2 = \frac{\sum_{i=1}^n r_{ik} x_{ij}^2}{\sum_{i=1}^n r_{ik}} - \mu_{kj}^2$$

Besides, we store r_{ik} as its log. (log probability) to avoid overflow

so: step 3:

$$r_{ik} = \log P_\theta(z_i = j, x_i)$$
$$= \log \pi_k |S_k|^{-\frac{1}{2}} e^{-\frac{1}{2}(x_i-\mu_k)^T S_k^{-1} (x_i - \mu_k)}$$
$$= \log \pi_k - \frac{1}{2} \log \prod_{j=1}^d S_{kj} - \frac{1}{2}(x_i - \mu_k)^T S_k^{-1} (x_i - \mu_k)$$
$$= \log \pi_k - \frac{1}{2} \sum_{j=1}^d \log S_{kj} - \frac{1}{2} \sum_{j=1}^d \frac{1}{S_{kj}} (x_{ij} - \mu_{kj})^2$$

step 4:

$$r_i = \log \sum_{k=1}^K P_\theta(z_i = j, x_i) = \log \sum_{k=1}^K e^{r_{ik}} \quad \rightarrow \text{use logsumexp}$$
$$= \log P_\theta(x_i)$$

step 5:

$$r_{ik} = \log \frac{P_\theta(z_i = j, x_i)}{P_\theta(x_i)} = \log P_\theta(z_i = j, x_i) - \log P_\theta(x_i) = r_{ik} - r_i.$$

step 6:

$$l(\text{iter}) = -\sum_{i=1}^n \log P_\theta(x_i) = -\sum_{i=1}^n r_i.$$

From step 8, we can now restore $r_{ik} = e^{r_{ik}}$ because we have normalized it, so it won't overflow.

update of π_j doesn't change because it is not relavent to the dimention.
update of μ and variance follows derivation above:

$$\mu_{kj} = \frac{\sum_{i=1}^n r_{ik} x_{ij}}{\sum_{i=1}^n r_{ik}} \Rightarrow \mu_k = \frac{\sum_{i=1}^n r_{ik} x_i}{r_{\cdot k}}$$

$$S_{kj} = \frac{\sum_{i=1}^n r_{ik} x_{ij}^2}{\sum_{i=1}^n r_{ik}} - \mu_{kj}^2 \Rightarrow S_k = \frac{\sum_{i=1}^n r_{ik} x_i}{r_{k\cdot}} - \mu_k^2$$

Specifically, since S is a diagnol matrix, we only store the diagnol as a vector. The square μ_k^2 is element-wise.

Space complexity: $O(nd + kd + nk)$ as I will store μ and S as a $K \times d$ matrix, r as a $n \times k$ matrix, and the data is stored as $n \times d$ matrix. All other values are either smaller (as one vector) or the same.

Time complexity: $O(nkd)$

update of $\log r_{ik}$ take $O(k \times n \times d)$, $n \times d$ introduced by matrix productions of $(x-\mu)^2 \cdot \frac{1}{S}$ elementwise square and inverse.

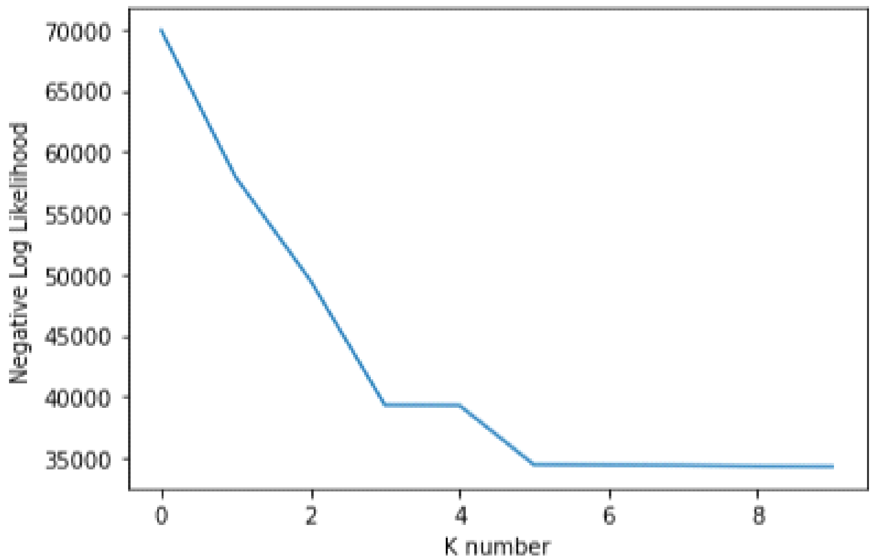
update of $\log r_{nk}$ take $O(k \times n \times d)$, $n \times d$ introduced by matrix production of $(X-\mu)^2 \cdot \frac{1}{S_k}$, elementwise square and inverse.

Normalization and loss and π updates only do subtraction or scalar multiplication, they cost $O(nk)$, $O(n)$, $O(k)$ respectively.

The update of μ and S requires matrix multiplication of $\mathbb{R}^{k \times n}$ and $\mathbb{R}^{n \times d}$, so they cost $O(nkd)$

$\Rightarrow$ total $O(nkd)$

Graph of negative likelihood v.s. K number.



I think the most appropriate value of K is number 5. The loss drops significantly before k=5 and almost does not change after k=5. This means that when K>=5, we almost have equally best GMM models. As we increase the number of components, we are like splitting some existing components into smaller ones with different weights. Since the components already have great modeling, splitting one component into more components with different weights is not really helpful and essentially is like using a "smaller GMM" to model that one component. It will cost more since the time complexity it related to the number of k when training but not a significant performance increase.

Model reports:

GMM Model with K = 5

```
========Parameters for model 0=========
---Weights = 0.012149022366402182
---Means = [-0.71118778 -0.51652233 -0.694506   -1.73751139  0.10280086 -0.65190711
 -2.0643607   0.24881163 -0.8323711   0.22952811  1.17827208 -0.14248053
  1.22096289 -0.89979927  1.05089612 -0.4901747   0.05149158 -0.87880974
  0.08197796  0.03603793]
---Variance = [0.64627987 2.35182502 1.00524562 0.13439717 2.46732699 0.85261726
 1.12499833 2.27824505 1.0313166  0.44311031 0.86006935 0.5709815
 0.91348845 0.43266023 1.12378261 0.80544136 0.94170524 0.53833318
 0.53379455 0.0370334 ]
========Parameters for model 1=========
---Weights = 0.19966381575964276
---Means = [-0.44267848  0.50391154  0.950766    0.81728867  2.09622268 -1.20176037
  0.47390156 -0.18721435  0.71796721  0.90632485  0.08468847  0.90059829
 -0.05950644 -0.93576893  0.05206341  0.25547972  1.34236937  0.50456567
```

-0.07727331 -0.58232248]
---Variance = [5.55600904 1.14899773 0.84355382 3.45094378 0.97609881 1.14948736
0.56444693 1.23614187 1.93669948 1.28180856 1.20297403 0.96079463
1.33795462 2.08707011 0.60242178 0.77659495 0.79565871 0.67950758
0.6038577 1.04178272]
========Parameters for model 2=========
---Weights = 0.2000568641754918
---Means = [-1.04245507 -1.3932431 -1.70825617 1.91688398 -0.5408255 -0.44208073
-1.27445681 0.76625967 -1.57571943 -0.22032739 -0.89487853 0.38816372
-0.5371024 -1.16517172 -0.04043416 0.44105374 0.04694923 0.30218401
-0.65274396 -0.34949195]
---Variance = [1.60218958 0.45551774 0.16937255 0.613258   2.49069457 0.88631499
0.87429067 1.15515943 1.38452757 0.466079   0.06871052 1.75239497
0.75036747 0.79986686 0.10019445 0.67225357 1.12720575 1.03094454
1.13974006 0.43648473]
========Parameters for model 3=========
---Weights = 0.28790691759115294
---Means = [-0.66237454 -0.39338161 -0.84402568 -1.71676513 0.19299571 -0.35905465
-1.62649126 0.48927106 -0.9496469   0.02513854 0.73050435 0.10147877
1.14519113 -1.22377676 0.41055964 -0.71143272 -0.93308894 -0.55724564
-0.33668757 0.05624604]
---Variance = [0.49224444 1.86735304 0.98852854 0.08013825 1.16299244 0.81724326
0.9688641 1.51314211 1.11781767 0.30994037 0.86592261 0.3311817
0.83167828 0.68635855 1.05053095 0.66681669 0.71357832 0.68711859
0.43528422 0.01683952]
========Parameters for model 4=========
---Weights = 0.3002233801073085
---Means = [-1.14978779 0.93516206 0.46951253 -1.54739486 1.50788616 1.92709509
1.20168114 -0.18015472 -1.0567946   1.08925075 -0.33255357 1.2268286
0.21180603 0.97292598 0.3526964   0.7222799   0.0106627   1.8061595
0.09292342 0.39997164]
---Variance = [0.35429502 1.3040312 0.66527067 2.26419042 0.627489   1.58079312
1.12668034 0.04878143 0.75463334 1.64247456 1.2876397 0.28287609
0.04109251 1.1009087 0.5115924 0.16765842 0.78528318 0.77080565
2.1333621 1.35676591]

B)

The error rate as a function of K is as follows:

When K = 0, Error rate: 0.1255
When K = 1, Error rate: 0.1126
When K = 2, Error rate: 0.0987
When K = 3, Error rate: 0.0895
When K = 4, Error rate: 0.0847
When K = 5, Error rate: 0.0763
When K = 6, Error rate: 0.0717
When K = 7, Error rate: 0.0779
When K = 8, Error rate: 0.072
When K = 9, Error rate: 0.0698