CSE 6740 HW1

Che-Ting,Meng

September 6, 2022

1 Probability

1.1

$p = \frac{e^{-n}}{2^n}$

1.2

P (infected|files corrupted)=$\frac{0.15*0.95*0.2}{0.15*0.95*0.2+0.85*0.1*0.2}$=0.626

1.3

1.3.1 (a)

P $(Walk_n)$=$\begin{cases} p & \text{for } n = 1; \\ \frac{p}{3^{n-1}} + \frac{1}{2}\sum \frac{1}{3n-1} & \text{for } n > 1. \end{cases}$

1.3.2 (b)

P $(Late_n)$=$\frac{p}{3^n} + \frac{1}{2}\sum \frac{1}{3^n}$

2 Maximum Likelihood

2.1 (a)

$l(\beta; x) = log \sqcap \frac{1}{\beta} e^{\frac{-x}{\beta}} = -n * \log \beta - \frac{\log e}{\beta} \sum x$

$$\begin{aligned} \frac{\partial l}{\partial \beta} &= -\frac{n}{\beta} + \frac{\log e \sum x}{\beta^2} = 0 \\ \beta_{MLE} &= \frac{\log e \sum x}{n} \end{aligned} \tag{1}$$

2.2 (b)

$l(x_0, \theta; x) = \log \sqcap \theta x_0^{\theta} x^{-\theta-1} = n \log \theta + n\theta \log x_0 - (\theta + 1) \sum \log x$

$$\begin{aligned} \frac{\partial l}{\partial \theta} &= \frac{n}{\theta} + n \log x_0 - \sum \log x = 0 \\ \theta_{MLE} &= \frac{n}{\sum \log x - n \log x_0} \end{aligned} \tag{2}$$

2.3 (c)

$$\begin{aligned} l(\beta, \sigma^2; y, X) &= log[\sqcap (2\pi\sigma^2)^{-\frac{N}{2}} \exp(-\frac{1}{2\sigma^2}) \sum (y_i - x_i\beta)^2] \\ &= -\frac{N}{2}(n \log 2\pi) - \frac{N}{2} N \log \sigma^2 - N \frac{\sum (y_i - x_i\beta)^2}{2\sigma^2} \end{aligned}$$

$$\begin{aligned} \hat{\beta_N} &= -N \frac{\sum (y_i - x_i\beta)^2}{2\sigma^2} = 0 \\ \frac{N}{2\sigma^2} * \sum 2 * (-x_i)(y_i - x_i\beta) &= 0 \\ \sum 2 * (-x_i)(y_i - x_i\beta) &= 0 \\ \sum (x_i y_i) &= \sum (x_i^2 \beta) \\ (X^T X) * \beta &= X^T y \\ \beta &= (X^T X)^{-1} X^T y \end{aligned} \tag{3}$$

$$\hat{\sigma_N^2} = \frac{\partial L}{\partial \sigma^2} = -\frac{N^2}{2\sigma^2} + \frac{N\sum (y_i - x_i)^2}{2\sigma^4} = 0$$
$$\sum (y_i - x_i)^2 = N\sigma^2 \tag{4}$$
$$\sigma^2 = \frac{1}{N}\sum (y_i - x_i)^2$$

3 PCA

3.1

$$\mathrm{w}=\begin{Bmatrix} \overrightarrow{w_1} \\ \overrightarrow{w_2} \\ ... \\ \overrightarrow{w_q} \end{Bmatrix} \; w^T=\left\{ \overrightarrow{\mathrm{w}_1^T}, \overrightarrow{\mathrm{w}_2^T}, ..., \overrightarrow{\mathrm{w}_q^T} \right\}$$

For length-one vectors $\overrightarrow{w}_i * \overrightarrow{w}_i = 1$ and for orthogonal vectors $\overrightarrow{w}_i * \overrightarrow{w}_j = 0$ (all $i \neq j$)

$$w * w^T = \begin{Bmatrix} \overrightarrow{w_1} * \overrightarrow{w_1}, \overrightarrow{w_1} * \overrightarrow{w_2}, ..., \overrightarrow{w_1} * \overrightarrow{w_q} \\ \overrightarrow{w_2} * \overrightarrow{w_1}, \overrightarrow{w_2} * \overrightarrow{w_2}, ..., \overrightarrow{w_2} * \overrightarrow{w_q} \\ ... \\ \overrightarrow{w_q} * \overrightarrow{w_1}, \overrightarrow{w_q} * \overrightarrow{w_2}, ..., \overrightarrow{w_q} * \overrightarrow{w_q} \end{Bmatrix} = \begin{Bmatrix} 1,0,...,0 \\ 0,1,...,0 \\ ... \\ 0,0,...,1 \end{Bmatrix} = I_q$$

3.2

for projection x_i':

$x_i' = \frac{w^T x_i}{w^T w} w = (w^T x_i) w$

3.3

for $x_i \in datasetD$;$\mathrm{x}_i' = (w^T x_i)w$; $\epsilon_i = x_i - x_i'$ along direction w:

$$
\begin{aligned}
MSE(w) &= \frac{1}{n}\sum ||\epsilon_i||^2 \\
&= \frac{1}{n}\sum ||x_i - x_i'||^2 \\
&= \frac{1}{n}\sum (x_i - x_i')^T(x_i - x_i') \\
&= \frac{1}{n}\sum (||x_i||^2 - 2x_i^T x_i' + (x_i')^T x_i') \\
&= \frac{1}{n}\sum (||x_i||^2 - 2x_i^T(w^T x_i)w + ((w^T x_i)w)^T(w^T x_i)w) \\
&(from x_i' = (w^T x_i)w) \\
&= \frac{1}{n}\sum (||x_i||^2 - 2(w^T x_i)(x_i^T w) + (w^T x_i)(x_i^T w)w^T w) \\
&= \frac{1}{n}\sum (||x_i||^2 - (w^T x_i)(x_i^T w)) \\
z &= \frac{1}{n}\sum (||x_i||^2) - w^T(\frac{1}{n}\sum (x_i x_i^T)w) \\
&(\lambda x_i : eigenvectors) \\
&= \frac{1}{n}\sum (||x_i||^2) - \lambda x_i
\end{aligned}
$$

3.4

4 Clustering

4.1

Clustering centers at set(1, 3, 6.5).

4.2

Sub-optimal initialization: initialize centers at(5, 6, 7), clustering centers would end at set(2, 6, 7).

4.3

The adjustment step decreases objective:

$$c^j = \frac{1}{|i:\pi(i)=j|} \sum_{i:\pi(i)=j} x^i = argmin_c \sum_{i:\pi(i)=j} ||x^i - c||^2$$

4.4

Euclidean distance can only recognize the absolute distance between data points, therefore, can only compute linear boundaries and cannot handle high dimensional data.
A good solution would be using spectral clustering where we use adjacency matrix to compute a graph Laplacian, use eigen decomposition to have eigenvalues correspond to the number of component connections and eigenvectors to cluster assignment, then run k means on eigenvectors as new data points.
The benefits of spectral clustering is that it accounts for the connection between data points, making it possible to compute non-linear clusters.

5 Programming: Image Compression

5.1

For my K-medoids, I applied the following implementations:
1) initialize the representatives by binning K intervals from 0 to the maximum of the RGB value within all pixels of the picture given;
2) apply the Manhattan distance measure;
3) terminating when total summation of distance is the same between two adjoining loops.

5.2

Picture size 320x240.

5.3

I ran my K-mediods with K-values 5, 18 and 33. With larger K-values:
1) the image is clearer and less compressed because there are more RGB combinations available;
2) the run time difference significantly increases as the K-value increases. It took 7sec, 16sec, 49sec respectively to run K-mediods-5,18,33;
3) the occurrences of empty clusters increases.

Results are shown in section[5.5].

5.4

Random initialization does affect the final result, this is because the K-mediods algorithm is not convex and converged to local optimums.
On the other hand, with the same initial locations, the result remains the same. Because the distance measure would give out the same calculations and centroid assignments for the same initial locations, then converge at the same place.

5.5

There are slight differences in K-values because I applied fixated initial representatives to make sure the comparison is under controlled variables, and for some K-values there are empty clusters. I adopted K-values without empty clusters as closely as possible.

5.5.1 K-means-5 vs. K-medoids-5

K-means-5

K-medoids-5

5.5.2 K-means-16 vs. K-medoids-18

K-means-16

K-medoids-18

5.5.3 K-means-32 vs. K-medoids-33

K-means-32

K-medoids-33

5.5.4 Conclusions

1) K-means at O(n*k*i) complexity is significantly faster than K-mediods at O($n^2 * k * i$) complexity.
2) K-mediods with Manhattan distance measure is more robust because it is less sensitive to outliers and noise.
3) In this case, K-mediods produces more compressed images, so it provides better output quality under the same K-values.