CSE 6740 - HW 1

Manvitha Kalicheti
gtID: 903838438

September 21, 2022

Collaborators

Keerthan Ramnath, Aishwarya Vijaykumar Sheelvant

References

Pattern Recognition and Machine Learning by Christopher Bishop, Class slides
For debugging and finding different methods: geeksforgeeks.org, stackoverflow.com, towardsdatascience.com

1

1.

$$\begin{aligned} P(heads) &= \sum_{i=1}^{\infty} \frac{1}{2^n} e^{-n} \\ &= \frac{1}{2e}\left(1 + \frac{1}{2e} + \frac{1}{(2e)^2} + ...\right) \\ &= \frac{1}{2e}\left(\frac{1}{1 - \frac{1}{2e}}\right) \\ &= \boxed{\frac{1}{2e-1}} \end{aligned}$$

2.

$$\begin{aligned} P(V) =& \text{Probability of a computer being actually infected with virus V} \\ P(C) =& \text{Probability of a computer having corrupted files} \\ P(V|C) =& \text{Probability of a computer being actually infected with virus V,} \\ & \text{given that the computer has corrupted files} \\ P(C|V) =& \text{Probability of a computer having corrupted files,} \\ & \text{given that the computer is actually infected with virus V} \end{aligned}$$

$$
\begin{aligned}
P(V|C) &= \frac{P(C|V)P(V)}{P(C)} \\
&= \frac{(0.95)(0.2)(0.15)}{(0.95)(0.2)(0.15) + (0.85)(0.2)(0.1)} \\
&= \boxed{0.6264}
\end{aligned}
$$

3.

(a)

$$
\begin{aligned}
P(w, k) =& \text{Probability of Charlie walking on k-th day} \\
P(b, k) = 1 - P(w, k) =& \text{Probability of Charlie taking the bus on k-th day} \\
P(t(\text{or } l)|w(\text{or } b)) =& \text{Probability of Charlie being on time (or being late),} \\
& \text{given that Charlie walked (or took the bus)}
\end{aligned}
$$

$$
\begin{aligned}
P(w, n) &= P(w, n-1)P(t, w) + P(b, n-1)P(l, b) \\
\text{Let } P(w, k) &= S_k \\
S_n &= S_{n-1}\frac{1}{2} + (1 - S_{n-1})\frac{1}{6} \\
S_n &= \frac{S_{n-1}}{3} + \frac{1}{6} \\
||ly, S_{n-1} = \frac{S_{n-2}}{3} + \frac{1}{6} & \\
S_n &= \frac{\frac{S_{n-2}}{3} + \frac{1}{6}}{3} + \frac{1}{6} \\
S_n &= \frac{S_{n-2}}{3^2} + \frac{1}{6.3} + \frac{1}{6} \\
\text{Again, } S_{n-2} = \frac{S_{n-3}}{3} + \frac{1}{6} & \\
S_n &= \frac{\frac{S_{n-3}}{3} + \frac{1}{6}}{3^2} + \frac{1}{6.3} + \frac{1}{6} \\
S_n &= \frac{S_{n-3}}{3^3} + \frac{1}{6.3^2} + \frac{1}{6.3} + \frac{1}{6} \\
\text{In general} & \\
S_n &= \frac{S_{n-k}}{3^k} + \frac{1}{6}\left(1 + \frac{1}{3} + ... + \frac{1}{3^{k-1}}\right) \\
\text{put } k = n-1 & \\
S_n &= \frac{S_{n-(n-1)}}{3^{n-1}} + \frac{1}{6}\left(1 + \frac{1}{3} + ... + \frac{1}{3^{n-2}}\right) \\
S_n &= \frac{S_1}{3^{n-1}} + \frac{1}{6}\left(\frac{1.(1 - \frac{1}{3}^{n-1})}{\frac{2}{3}}\right) \\
S_1 = P(w, 1) = p &
\end{aligned}
$$

$$S_n = \boxed{P(w,n) = \left(p - \frac{1}{4}\right)\left(\frac{1}{3}\right)^{n-1} + \frac{1}{4}}$$

(b)

$$\begin{aligned} P(\text{Charlie gets late on n-th day}) &= P(w,n)P(l,w) + P(b,n)P(l,b) \\ &= P(w,n)P(l,w) + (1 - P(w,n))P(l,b) \\ &= \left(\left(p - \frac{1}{4}\right)\left(\frac{1}{3}\right)^{n-1} + \frac{1}{4}\right)\frac{1}{2} + \left(1 - \left(p - \frac{1}{4}\right)\left(\frac{1}{3}\right)^{n-1} - \frac{1}{4}\right)\frac{1}{6} \\ &= \boxed{\left(p - \frac{1}{4}\right)\left(\frac{1}{3}\right)^{n} + \frac{1}{4}} \end{aligned}$$

2

(a)

$P(x|\beta) = \frac{1}{\beta}e^{-\frac{x}{\beta}}$

$D = \{x_1, x_2, ...x_n\}$

$$\begin{aligned} L(\theta|x_1, x_2...x_n) &= P(x_1|\beta)P(x_2|\beta)....P(x_n|\beta) \\ L(\theta|D) &= \left(\frac{1}{\beta}\right)^n exp\left(-\frac{\sum_{i=1}^n x_i}{\beta}\right) \end{aligned}$$

Applying ln on both sides

$$\implies \ln(L(\theta|D)) = n\ln\left(\frac{1}{\beta}\right) - \frac{\sum_{i=1}^n x_i}{\beta}$$

Setting derivative of this log function wrt β to 0.

$$\begin{aligned} \frac{\partial}{\partial\beta}\ln(L(\theta|D)) = \frac{n}{1/\beta}\left(-\frac{1}{\beta^2}\right) + \frac{\sum_{i=1}^n x_i}{\beta^2} &= 0 \\ \frac{\sum_{i=1}^n x_i}{\beta^2} &= \frac{n}{\beta} \\ \frac{1}{\beta}\left(\frac{\sum_{i=1}^n x_i}{\beta} - n\right) &= 0 \implies \boxed{\hat{\beta} = \frac{n}{\sum_{i=1}^n x_i} = \frac{1}{\bar{x}}} \end{aligned}$$

(b)

$f(x|x_o, \theta) = \theta x_o^\theta x^{-\theta-1}$ where $x \geq x_o$ and $\theta > 1$

$$
\begin{aligned}
L(\theta|x_1, x_2, ...x_n) &= f(x_1|x_o,\theta)f(x_2|x_o,\theta)...f(x_n|x_o,\theta) \\
&= \theta^n x_o^\theta \left(\Pi_{i=1} n x_i^{-\theta-1}\right)
\end{aligned}
$$

Applying ln on both sides

$$
\begin{aligned}
\ln(L(\theta|D)) &= n\ln\theta + n\theta\ln x_o + \Pi_{i=1}^n \ln x_i^{-\theta-1} \\
\ln(L(\theta|D)) &= n\ln\theta + n\theta\ln x_o - (1+\theta)\Pi_{i=1}^n \ln x_i
\end{aligned}
$$

Setting derivative of this log function wrt θ to 0.

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\ln(L(\theta|D)) = \frac{n}{\theta} + n\ln x_o - \sum_{i=1}^{n} \ln x_i = 0 \\
\frac{n}{\theta} + n\ln x_o = \sum_{i=1}^{n} \ln x_i \\
\frac{n}{\theta} = \sum_{i=1}^{n} \ln x_i - n\ln x_o \implies \boxed{\hat{\theta} = \frac{n}{\sum_{i=1}^{n} \ln x_i - n\ln x_o}}
\end{aligned}
$$

(c)

$L(\beta,\sigma^2;y,x) = (2\pi\sigma^2)^{-N/2} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - x_i\beta)^2\right)$

Applying ln on both sides

$\ln(L(\beta,\sigma^2;y,x)) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - x_i\beta)^2$

Setting derivative of this log function wrt β to 0.

$$
\begin{aligned}
\frac{\partial}{\partial\beta}(\ln(L(\beta,\sigma^2;y,x))) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i\beta)(-x_i) = 0 \\
\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i y_i - x_i^2\beta) = 0
\end{aligned}
$$

$$
\implies \boxed{\hat{\beta}_N = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} \equiv (X^T X)^{-1} X^T y}
$$

Setting derivative of this log function wrt σ^2 to 0.

$$
\begin{aligned}
\frac{\partial}{\partial\sigma^2}(\ln(L(\beta,\sigma^2;y,x))) = -\frac{N2\pi}{2.2\pi.\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{N}(y_i - x_i\beta)^2 = 0 \\
\frac{N}{2\sigma^2} = \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - x_i\beta)^2
\end{aligned}
$$

$$\implies \boxed{\hat{\sigma}_N^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - x_1\hat{\beta}_N)}$$

3

1.

$w = \begin{bmatrix} \overrightarrow{w_1} & \overrightarrow{w_2} & . & . & . & \overrightarrow{w_q} \end{bmatrix}$

$$w^T = \begin{bmatrix} \overrightarrow{w_1} \\ \overrightarrow{w_2} \\ . \\ . \\ . \\ \overrightarrow{w_q} \end{bmatrix}$$

$$w^T w = \begin{bmatrix} \overrightarrow{w_1}.\overrightarrow{w_1} & \overrightarrow{w_1}.\overrightarrow{w_2} & \overrightarrow{w_1}.\overrightarrow{w_3} & . & . \\ \overrightarrow{w_2}.\overrightarrow{w_1} & \overrightarrow{w_2}.\overrightarrow{w_2} & . & . & . \\ \overrightarrow{w_3}.\overrightarrow{w_1} & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & \overrightarrow{w_q}.\overrightarrow{w_q} \end{bmatrix}$$

$\overrightarrow{w_1}, ...\overrightarrow{w_q}$ are orthogonal $\implies \overrightarrow{w_i}.\overrightarrow{w_j} = 0$ if $i \neq j$.

$\overrightarrow{w_1}, ...\overrightarrow{w_q}$ are length one $\implies \overrightarrow{w_i}.\overrightarrow{w_j} = 1$ if $i = j$.

$$\therefore w^T w = \begin{bmatrix} 1 & 0 & 0 & . & . & . \\ 0 & 1 & 0 & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & 1 \end{bmatrix}$$

$\boxed{w^T w = I_q}$

2.

$\mathbf{x} = \begin{bmatrix} \overrightarrow{x_1} & \overrightarrow{x_2} & . & . & . & \overrightarrow{x_n} \end{bmatrix}$ where each x_i is a p-dimensional vector.

$dim(\mathbf{x}) = p$ x n

$\mathbf{w} = \begin{bmatrix} \overrightarrow{w_1} & \overrightarrow{w_2} & . & . & . & \overrightarrow{w_q} \end{bmatrix}$ where each w_i is a p-dimensional vector.

$dim(\mathbf{w}) = p$ x q

$\mathbf{x'} = \left((\mathbf{x}.\mathbf{w})\mathbf{w^T}\right)^T$

$\mathbf{x'} = \left((\mathbf{x^T}\mathbf{w})\mathbf{w^T}\right)^T$

$$\mathbf{x'} = \left(\left(\begin{bmatrix} \overrightarrow{x_1}^T \\ \overrightarrow{x_2}^T \\ . \\ . \\ . \\ \overrightarrow{x_n}^T \end{bmatrix}_{n \text{ x } p} . \begin{bmatrix} \overrightarrow{w_1} & \overrightarrow{w_2} & . & . & . & \overrightarrow{w_q} \end{bmatrix}_{p \text{ x } q} \right) \begin{bmatrix} \overrightarrow{w_1} \\ \overrightarrow{w_2} \\ . \\ . \\ . \\ \overrightarrow{w_q} \end{bmatrix} \right)^T$$

$$\mathbf{x'} = \left(\begin{bmatrix} \overrightarrow{x_1}.\overrightarrow{w_1} & \overrightarrow{x_1}.\overrightarrow{w_2} & . & . & . & \overrightarrow{x_1}.\overrightarrow{w_q} \\ \overrightarrow{x_2}.\overrightarrow{w_1} & \overrightarrow{x_2}.\overrightarrow{w_2} & . & . & . & \overrightarrow{x_2}.\overrightarrow{w_q} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \overrightarrow{x_n}.\overrightarrow{w_1} & \overrightarrow{x_n}.\overrightarrow{w_2} & . & . & . & \overrightarrow{x_n}.\overrightarrow{w_q} \end{bmatrix}_{n \text{ x } q} \begin{bmatrix} \overrightarrow{w_1} \\ \overrightarrow{w_2} \\ . \\ . \\ . \\ \overrightarrow{w_q} \end{bmatrix} \right)^T$$

$\mathbf{x'} = \begin{bmatrix} \sum_{k=1}^{q} (\overrightarrow{x_1}.\overrightarrow{w_k})\overrightarrow{w_k} & \sum_{k=1}^{q} (\overrightarrow{x_2}.\overrightarrow{w_k})\overrightarrow{w_k} & . & . & . & \sum_{k=1}^{q} (\overrightarrow{x_n}.\overrightarrow{w_k})\overrightarrow{w_k} \end{bmatrix}_{\text{p x n}}$

When $q = 1$, $\mathbf{x'} = \begin{bmatrix} (\overrightarrow{x_1}.\overrightarrow{w_1})\overrightarrow{w_1} & (\overrightarrow{x_2}.\overrightarrow{w_1})\overrightarrow{w_1} & . & . & . & (\overrightarrow{x_n}.\overrightarrow{w_1})\overrightarrow{w_1} \end{bmatrix}_{\text{p x n}}$

That is, $x'_i = (\overrightarrow{x_i}.\overrightarrow{w_1})\overrightarrow{w_1}$

3.

$$\begin{aligned} SSE &= ||\overrightarrow{x_i} - \sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})\overrightarrow{w_k}||^2 \\ &= \left(\overrightarrow{x_i} - \sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})\overrightarrow{w_k} \right) . \left(\overrightarrow{x_i} - \sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})\overrightarrow{w_k} \right) \\ &= (\overrightarrow{x_i}.\overrightarrow{x_i}) - 2\sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})(\overrightarrow{x_i}.\overrightarrow{w_k}) + \sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})^2 (\overrightarrow{w_k}.\overrightarrow{w_k}) \end{aligned}$$

From 1., $(\overrightarrow{w_k}.\overrightarrow{w_k}) = 1$

$$\begin{aligned} &= ||x_i||^2 - 2\sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})^2 + \sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})^2 \\ &= ||x_i||^2 - \sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})^2 \\ MSE &= \frac{SSE}{n} \\ MSE &= \underbrace{\frac{||x_i||^2}{n}}_{\text{term 1}} - \underbrace{\frac{1}{n} \sum_{k=1}^{q} (\overrightarrow{x_i}.\overrightarrow{w_k})^2}_{\text{term 2}} \end{aligned}$$

Clearly, term 1 only depends on X and term 2 only depends on the projections of X along the q directions.

4.

Minimising projection residuals $\equiv$ minimising MSE

$$MSE = \underbrace{\frac{||x_i||^2}{n}}_{\text{term 1}} - \underbrace{\frac{1}{n}\sum_{k=1}^{q}(\overrightarrow{x_i}.\overrightarrow{w_k})^2}_{\text{term 2}}$$

We know that $Var(\overrightarrow{x_i}.\overrightarrow{w_k}) = \frac{1}{n}\sum_{i=1}^{n}(\overrightarrow{x_i}.\overrightarrow{w_k})^2 - (\frac{1}{n}\sum_{i=1}^{n}\overrightarrow{x_i}.\overrightarrow{w_k})^2$

To minimise MSE, we have to maximise term 2. Consider term 2:

$\frac{1}{n}\sum_{k=1}^{q}(\overrightarrow{x_i}.\overrightarrow{w_k})^2 = \frac{1}{n}(\overrightarrow{x_i}.\overrightarrow{w_1}) + \frac{1}{n}(\overrightarrow{x_i}.\overrightarrow{w_2}) + ... + \frac{1}{n}(\overrightarrow{x_i}.\overrightarrow{w_q})$

Consider one of these terms:

for some $k \leq q$

$\frac{1}{n}\sum_{i=1}^{n}(\overrightarrow{x_i}.\overrightarrow{w_k})^2 = (\frac{1}{n}\sum_{i=1}^{n}\overrightarrow{x_i}\overrightarrow{w_k})^2 + Var(\overrightarrow{x_i}.\overrightarrow{w_k})$

We assume the data is centered, i.e., $\frac{1}{n}\sum_{i=1}^{n}\overrightarrow{x_i} = 0$

$\implies \frac{1}{n}\sum_{i=1}^{n}(\overrightarrow{x_i}.\overrightarrow{w_k})^2 = Var(\overrightarrow{x_i}.\overrightarrow{w_k})$\$

Term 2 of $MSE = \sum_{k=1}^{q} Var(\overrightarrow{x_i}.\overrightarrow{w_k})$

$\therefore$ minimising MSE $\equiv$ maximising term 2 of MSE $\equiv$ maximising sum of variances along the different directions.

4

1.

Let us randomly initialise the 3 centers first:

$\mu_1 = 1, \mu_2 = 4, \mu_3 = 8$

Then, for each point, we allot a cluster based on how close a point is to the cluster center. We get x_1 belongs to cluster 1, x_2 belongs to cluster 2, x_3 could belong to either cluster 2 or 3. Let's pick cluster 3. x_4 belongs to cluster 3.

We now compute the loss function:

$J = \sum_{j=1}^{k}\sum_{i=1}^{n} r_{ij}||x_i - \mu_j||^2$ where

$$r_{ij} = \begin{cases} 1 & \text{point } i \in \text{ cluster } j \\ 0 & \text{point } i \notin \text{ cluster } j \end{cases}$$

We get $J = 5$.

We recompute the new centers as the mean of the points in each clusters.

$\mu_1 = 1, \mu_2 = 4.5, \mu_3 = 7$

Again, we assign points to these centers.

x_1 belongs to cluster 1. x_2 belongs to cluster 2. x_3 belongs to cluster 3. x_4 belongs to cluster 4.

We calculate $J = 3.25$.

Again, we recompute centers for these clusters.

$\mu_1 = 1, \mu_2 = 3, \mu_3 = 7$

Again, we assign points to these centers.

x_1 belongs to cluster 1. x_2 belongs to cluster 2. x_3 belongs to cluster 3. x_4 belongs to cluster 3.

We see no changes in cluster assignment.

We calculate $J = 0.5$

Recomputing centers for these clusters gives us

$\mu_1 = 1, \mu_2 = 3, \mu_3 = 6.5$

Again, we assign points to these centers.

x_1 belongs to cluster 1. x_2 belongs to cluster 2. x_3 belongs to cluster 3. x_4 belongs to cluster 3.

Recomputing centers tells us that the centers haven't changed from last time. We reach the end of our algorithm with the following clusters.

Centers	$\mu_1 = 2$	$\mu_2 = 3$	$\mu_3 = 6.5$
Points	$\mathrm{x}_1 = 1$	$\mathrm{x}_2 = 3$	$\mathrm{x}_3 = 6, x_4 = 7$

2.

Consider the following center initialisation:

$\mu_1 = 1, \mu_2 = 6, \mu_3 = 7$

We get $x_1, x_2 \in C1, x_3 \in C2, x_4 \in C4$.

Cost $J = 4$.

Recomputing centers: $\mu_1 = 2, \mu_2 = 6, \mu_3 = 7$

$x_1, x_2 \in C1, x_3 \in C2, x_4 \in C4$

Cost $J = 2$.

Recomputing centers: $\mu_1 = 2, \mu_2 = 6, \mu_3 = 7$

Centers did not change $\implies$ these are our final centers with cost $J = 2$.

But from **1.**, we know that it is possible to have a set of cluster centers with even lesser cost of $J = 0.5$

Thus, we see that changing the center initialization may give a suboptimal cluster assignment that cannot be further improved.

3.

The cost function in kmeans is defined as $J = \sum_{j=1}^{k} \sum_{i=1}^{n} r_{ij} ||x_i - \mu_j||^2$ where

$$r_{ij} = \begin{cases} 1 & \text{point } i \in \text{ cluster } j \\ 0 & \text{point } i \notin \text{ cluster } j \end{cases}$$

Our goals is to minimise this cost. We perform two steps iteratively in kmeans.

The first one is the Expectation (E) step where data points are allotted to clusters. Here, for each data point, we pick the cluster center that is closest to it (Euclidean distance). The $\sum_{i=1}^{n}$ part of the cost function reduces every time we do this step.

The second one is the Maximisation (M) step where we recompute the cluster centers. To do this, we find the mean of all the data points in a cluster and make that the new cluster center. The $\sum_{j=1}^{k}$ part of the cost function reduces every time we do this step.

As both steps cause the cost function to monotonically decrease, it is clear that the algorithm will converge, although only to a local optimum.

There are only a finite number of possible partitions- $\binom{n}{k}$. In E step, clustering is based only on the previous clustering stage. If the clustering doesn't change in this step, it will never change again. We reach convergence. If the clustering changes, then the new cost will definitely be lower than the old cost. This implies that the kmeans algorithm will take a finite number of steps to converge to a local optimum.

4.

Using Euclidean distance as the distance measure restricts the type of data that can be used to perform kmeans clustering. For instance, we cannot find Euclidean distance for non numerical (say, categorical) data.

Also, Euclidean distance sometimes fails as a good metric to cluster based on the shape of the data. Kmeans works well only when the clusters are nearly spherical. Consider two annular, concentric clusters of data points. The euclidean distance between a point on one ring and a point on another ring might be small enough for them both to be allotted the same cluster, but that is not the clustering we're looking for! We want the two rings to be placed in two different clusters. This cannot be achieved with kmeans.

For data with complicated geometric shapes, kmeans using Euclidean distance measure fails. In these cases, we use kernel methods to help kmeans achieve proper clustering. Algorithms like spectral clustering work well in these cases.

5.

Report follows. Code submitted separately.