CS260 Homework1

Yang Yang 804522285

Oct 5, 2015

1 Sequence of Coin Flips

1.1 Part A

The function of $\mathbb{P}[X = k]$ as a function of p is,

$$\mathbb{P}[X = k] = (1 - p)^{(k-1)}p \tag{1.1}$$

1.2 Part B

To make sure $X \geq x_0$, in the first $x_0 - 1$ situation, we should all observing the tail. Thus the probability of $\mathbb{P}[X = k]$ should be the product of $(1 - p)$ in the first $(x_0 - 1)$situation, which is,

$$\mathbb{P}[X = k] = (1 - p)^{(k-1)} \tag{1.2}$$

1.3 Part C

Because p is uniformly distributed in the $[0, 1]$, we know that,

$$\mathbb{P}[p > 1/2] = \frac{1}{2} \tag{1.3}$$

We also know that $\mathbb{P}[x = 1|p > 1/2] = p$ and $\mathbb{P}[x = 1] = p$
Thus, from Bayes Rule, we have,

$$\mathbb{P}[p > 1/2|x = 1] = \frac{\mathbb{P}[x = 1|p > 1/2]\mathbb{P}[p > 1/2]}{\mathbb{P}[x = 1]} = \frac{1}{2} \tag{1.4}$$

Therefore, observing the head in the first flip does not affect the probability of the event$\{p > 1/2\}$

2 Convex Functions and Information Theory

2.1 Part A

1° Consider $f_1(x) = |x|$ first.
Using the definition of the convex function, we have,

$$\begin{aligned} f_1(\theta x + (1 - \theta)y) &= |\theta x + (1 - \theta)y| \\ &\leq \theta|x| + (1 - \theta)|y| = \theta f_1(x) + (1 - \theta)f_1(y) \end{aligned} \tag{2.1}$$

Thus, $f_1(x) = |x|$ is a convex function.
2° Then Consider $f_2(x) = \exp(x)$.
By taking the second-order derivative of the function, we have,

$$f_2''(x) = \exp(x) > 0 \tag{2.2}$$

Thus, $f_2(x)$ is a convex function.
As the sum of two convex functions is also a convex function, we can say $f(x) = f_1(x) + f_2(x)$ is a convex function.

2.2 Part B

The entropy is defined as,

$$H(x) = -\sum_{i=1}^{k} p_i \log p_i \tag{2.3}$$

And we also have the constrains,

$$\sum_{i=1}^{k} p_i = 1 \tag{2.4}$$

Using Lagrange multiplier, we have,

$$\mathcal{L} = -\sum_{i=1}^{k} p_i \log p_i + \lambda \left(\sum_{i=1}^{k} p_i - 1 \right) \quad (\lambda \geq 0) \tag{2.5}$$

To maximize the above equation, taking derivative with respect to p_i,

$$\frac{\partial \mathcal{L}}{\partial p_i} = -1 - \log p_i + \lambda = 0 \rightarrow p_i = \exp(\lambda - 1) \tag{2.6}$$

By using $\sum_{i=1}^{k} p_i = 1$,

$$k \times \exp(\lambda - 1) = 1 \rightarrow \lambda = \log \frac{1}{k} + 1 \tag{2.7}$$

Thus,

$$p_i = \exp(\lambda - 1) = \frac{1}{k} \tag{2.8}$$

3 Linear Algebra

3.1 Part A

$\boldsymbol{\Sigma}$ is positive semi-definite.

Proof: Suppose there is a non-random vector $\boldsymbol{v} \in \mathbb{R}^d$, which d is equal to the dimension of the random vector $\boldsymbol{X}$. What we need to show is $\boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v} \geq 0$.

Using the expression of the covariance matrix, we have,

$$\boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v} = \boldsymbol{v}^T \mathbb{E}[(\boldsymbol{X} - \mathbb{E}\boldsymbol{X})(\boldsymbol{X} - \mathbb{E}\boldsymbol{X})^T] \boldsymbol{v} = \mathrm{Var}(\boldsymbol{v}^T \boldsymbol{X}) \tag{3.1}$$

As $\boldsymbol{v}^T \boldsymbol{X} = \sum_{i=1}^{d} v_i X_i$ is univariate, thus,

$$\mathrm{Var}(\boldsymbol{v}^T \boldsymbol{X}) \geq 0 \rightarrow \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v} \geq 0 \tag{3.2}$$

So $\boldsymbol{\Sigma}$ is positive semi-definite.

3.2 Part B

The eigenvalues and eigenvectors are:

- $C = A + B$

  For C, the eigenvectors will still be $u_1, u_2, ..., u_D$

  The eigenvalue will be $\alpha_1 + \beta_1, \alpha_2 + \beta_2, ..., \alpha_D + \beta_D$

- $D = A - B$

  For D, the eigenvectors will still be $u_1, u_2, ..., u_D$

  The eigenvalue will be $\alpha_1 - \beta_1, \alpha_2 - \beta_2, ..., \alpha_D - \beta_D$

- $E = AB$

  For E, the eigenvectors will be $u_1^2, u_2^2, ..., u_D^2$

  The eigenvalue will be $\alpha_1\beta_1, \alpha_2\beta_2, ..., \alpha_D\beta_D$

- $F = A^{-1}B$

  For F, the eigenvectors will be $u_1^2, u_2^2, ..., u_D^2$

  The eigenvalue will be $\frac{\beta_1}{\alpha_1}, \frac{\beta_2}{\alpha_2}, ..., \frac{\beta_D}{\alpha_D}$

4 KNN Classification in MATLAB/Octave

4.1 Part C

The train and validation accuracy are shown in the table below. From the table we know that, when $k = 9$ we get the highest validation accuracy.

When $k = 9$, we can find the test accuracy is 0.8946.

k	1	3	5	7	9	11	13	15
Training Accuracy	0.7779	0.8316	0.8663	0.8842	0.8863	0.8905	0.8842	0.8705
Validation Accuracy	0.7558	0.8046	0.8329	0.8406	0.8689	0.8638	0.8560	0.8278
k	17	19	21	23				
Training Accuracy	0.8589	0.8526	0.8537	0.8453				
Validation Accuracy	0.8252	0.8226	0.8098	0.8252				

4.2 Part D

The decision boundaries with different k are shown in the pictures below. From these picture, we can see, obviously, with the increase of k, the decision will become smoother. The larger the k is, the smoother the decision will be.

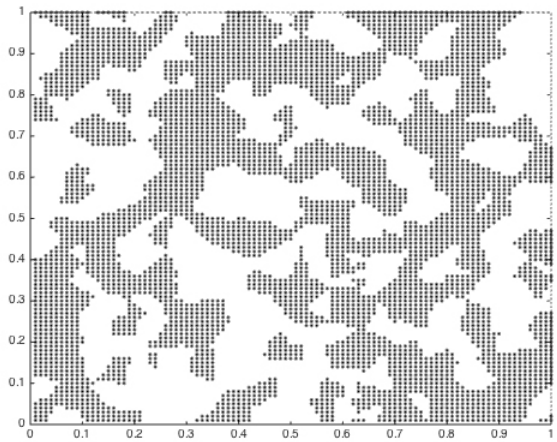

Figure 1 Boundary when $k = 1$

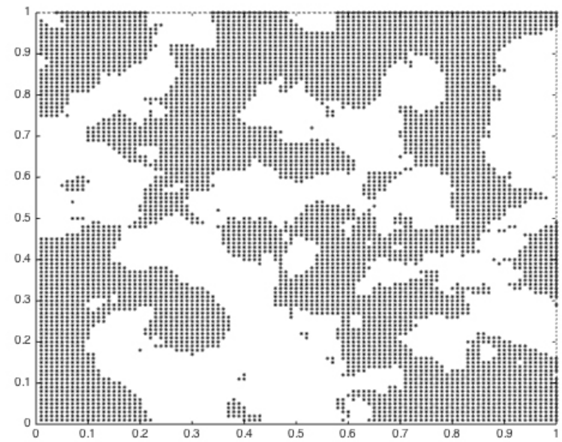

Figure 2 Boundary when $k = 5$

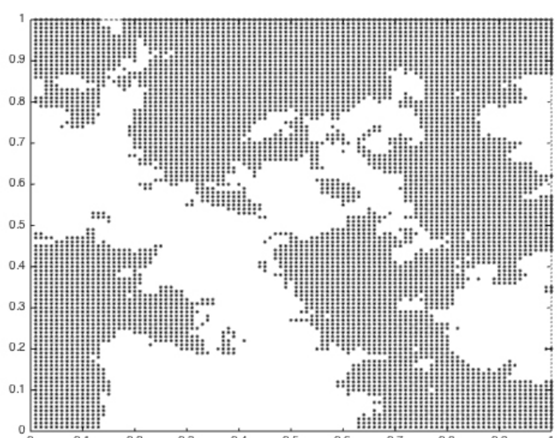

Figure 3 Boundary when $k = 15$

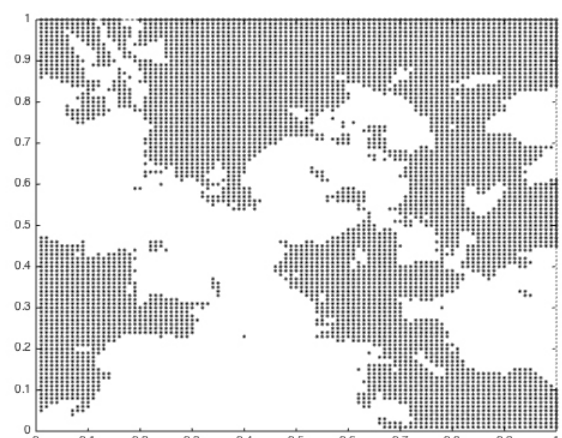

Figure 4 Boundary when $k = 20$