# ECE M146 - HW 5

May 14, 2020

John Rapp Farnes | 405461225

## 1

By the definition of Kernel, we have that $K(x, y) = \varphi(x)^T \varphi(y)$ for some function $\varphi$. As such

$$
\begin{aligned}
&K(x_1, x_2)^2 - K(x_1, x_1)K(x_2, x_2) \\
&= \underbrace{\varphi(x_1)^T \varphi(x_2)}_{=(\varphi(x_1)^T \varphi(x_2))^T} \cdot \varphi(x_1)^T \varphi(x_2) - \varphi(x_1)^T \varphi(x_1) \cdot \varphi(x_2)^T \varphi(x_2) \\
&= |\varphi(x_1)^T \varphi(x_2)|^2 - \|\varphi(x_1)\|^2 \|\varphi(x_2)\|^2
\end{aligned}
$$

By the Cauchy-Schwartz inequality we have that

$$|u^T v|^2 - \|u\|^2 \|v\|^2 \leq 0$$

for any vectors $u$ and $v$. Letting $u = \varphi(x_1)$ and $v = \varphi(x_2)$, we have that

$$K(x_1, x_2)^2 \leq K(x_1, x_1)K(x_2, x_2)$$

## 2

We have that $K$ valid kernel $\Leftrightarrow$ $K$ positive semi-definite $\Leftrightarrow$ $x^T K x \geq 0 \, \forall x \neq 0$

### 2.1 (a)

Hence, we have that

$$\begin{aligned} x^T K(x,x')x &= x^T(K_1(x,x') + K_2(x,x'))x \\ &= \underbrace{x^T K_1(x,x')x}_{\geq 0} + \underbrace{x^T K_2(x,x')x}_{\geq 0} \\ &\geq 0 \implies K \text{ valid kernel} \end{aligned}$$

### 2.2 (b)

We have that

$$\begin{aligned} x^T K x &= x^T (K_1 K_2) x \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} x_i \varphi_1(x_i)^T \varphi_1(x_j) \varphi_2(x_i)^T \varphi_2(x_j) x_j \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} x_i \sum_{k=1}^{N} [\varphi_1^{(k)}(x_i) \varphi_1^{(k)}(x_j) \varphi_2^{(k)}(x_i) \varphi_2^{(k)}(x_j)] x_j \\ &= \sum_{k=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} x_i \varphi_1^{(k)}(x_i) \varphi_1^{(k)}(x_j) \varphi_2^{(k)}(x_i) \varphi_2^{(k)}(x_j) x_j \\ &= \sum_{k=1}^{N} \sum_{i=1}^{N} \underbrace{[x_i \varphi_1^{(k)}(x_i) \varphi_2^{(k)}(x_i)]^2}_{\geq 0} \\ &\geq 0 \implies K \text{ valid kernel} \end{aligned}$$

### 2.3 (c)

For a given matrix $X$, we have $\exp(X) = \sum_{k=0}^{\infty} \frac{1}{k!} X^k$

Further we have that any scalar $a \geq 0$ is positive semi-definite, as

$$x^T a x = a x^T x = \underbrace{a}_{\geq 0} \underbrace{|x|^2}_{\geq 0} \geq 0$$

As such, $exp(K_1)$ is a sum of matrices $\frac{1}{k!} K_1^k$, which are products of positive semi-definite matrices (as $\frac{1}{k!} \geq 0 \, \forall k \in \mathbb{Z}$ and $K_1$ is PSD), and is therefore positive semi-definite.

## 3

By solving $\frac{\partial L}{\partial w} = 0$ when constructing the dual problem (lecture 8, the "soft" term does not change the result), we obtained

$$w = \sum_{m \in S} \alpha_m y^{(m)} x^{(m)}$$

We know from the problem formulation that vectors $x^{(n)}$ on the margin satisfy $w^T x^{(n)} + b = \pm 1$, depending on their label. The points on the margin are those with $\epsilon_n = 0$ and $\alpha_n > 0$. By $\beta_n \epsilon_n = 0$, we have $\beta_n > 0$ at such a point, and we therefore have by $C - \alpha_n - \beta_n = 0$ that $0 < \alpha_n < C \implies n \in M$. As such, we have:

$$w^T x^{(n)} + b = y^{(n)} \Leftrightarrow b = y^{(n)} - w^T x^{(n)} \forall n \in M$$

As shown in lecture, rather than obtaining $b$ for a single vector, a more numerically stable result is achieved by computing $b$ as the average of these equations, as such:

$$\begin{aligned} b &= \frac{1}{N_M} \sum_{n \in M} [y^{(n)} - w^T x^{(n)}] \\ &= \frac{1}{N_M} \sum_{n \in M} [y^{(n)} - (\sum_{m \in S} \alpha_m y^{(m)} x^{(m)})^T x^{(n)}] \\ &= \frac{1}{N_M} \sum_{n \in M} [y^{(n)} - \sum_{m \in S} \alpha_m y^{(m)} \langle x^{(n)}, x^{(m)} \rangle] \end{aligned}$$

# 4

## 4.1 (a)

We have, using Bayes' rule:

$$P(A = k|C = 0) = \frac{P(A = k, C = 0)}{P(C = 0)}$$

I will calculate these probabilities in python:

### 4.1.1 $P(A|C = 0)$

[1]:
```python
P_C_0 = .096+.024+.224+.056

P_A_0C_0 = .096+.024
P_A_1C_0 = .224+.056

P_A_0_C_0 = P_A_0C_0 / P_C_0
P_A_1_C_0 = P_A_1C_0 / P_C_0

print(f'P_A_0_C_0={P_A_0_C_0}')
print(f'P_A_1_C_0={P_A_1_C_0}')
```

```
P_A_0_C_0=0.3
P_A_1_C_0=0.7000000000000002
```

$$\therefore P(A = 0|C = 0) = 0.3, P(A = 1|C = 0) = 0.7$$

### 4.1.2 $P(B|C = 0)$

We can use the same formula as for $P(A|C = 0)$

[2]:
```python
P_B_0C_0 = .096+.224
P_B_1C_0 = .024+.056

P_B_0_C_0 = P_B_0C_0 / P_C_0
P_B_1_C_0 = P_B_1C_0 / P_C_0

print(f'P_B_0_C_0={P_B_0_C_0}')
print(f'P_B_1_C_0={P_B_1_C_0}')
```

```
P_B_0_C_0=0.8
P_B_1_C_0=0.2
```

$$\therefore P(B = 0|C = 0) = 0.8, P(B = 1|C = 0) = 0.2$$

#### 4.1.3 $P(A, B|C = 0)$

We have:

$$
\begin{aligned}
&P(A = i, B = j|C = 0) \\
&= P(A = i|C = 0)P(B = j|A = i, C = 0) \\
&= P(A = i|C = 0)\frac{P(A = i, B = j, C = 0)}{P(A = i, C = 0)}
\end{aligned}
$$

Which gives us four combinations $i \in \{0, 1\}, j \in \{0, 1\}$

[3]:
```
P_A_0B_0C_0 = .096
P_A_0_B_0_C_0 = P_A_0_C_0 * P_A_0B_0C_0 / P_A_0C_0

P_A_1B_0C_0 = .224
P_A_1_B_0_C_0 = P_A_1_C_0 * P_A_1B_0C_0 / P_A_1C_0

P_A_0B_1C_0 = .024
P_A_0_B_1_C_0 = P_A_0_C_0 * P_A_0B_1C_0 / P_A_0C_0

P_A_1B_1C_0 = .056
P_A_1_B_1_C_0 = P_A_1_C_0 * P_A_1B_1C_0 / P_A_1C_0

print(f'P_A_0_B_0_C_0={P_A_0_B_0_C_0}')
print(f'P_A_1_B_0_C_0={P_A_1_B_0_C_0}')
print(f'P_A_0_B_1_C_0={P_A_0_B_1_C_0}')
print(f'P_A_1_B_1_C_0={P_A_1_B_1_C_0}')
```

```
P_A_0_B_0_C_0=0.24
P_A_1_B_0_C_0=0.5600000000000002
P_A_0_B_1_C_0=0.06
P_A_1_B_1_C_0=0.14000000000000004
```

$$
\therefore
\begin{aligned}
P(A = 0, B = 0|C = 0) &= 0.24 \\
P(A = 1, B = 0|C = 0) &= 0.56 \\
P(A = 0, B = 1|C = 0) &= 0.06 \\
P(A = 1, B = 1|C = 0) &= 0.14
\end{aligned}
$$

## 4.2 (b)

### 4.2.1 $P(A|C = 1)$

[4]:
```
P_C_1 = .27+.03+.27+.03

P_A_0C_1 = .27+.03
P_A_1C_1 = .27+.03

P_A_0_C_1 = P_A_0C_1 / P_C_1
P_A_1_C_1 = P_A_1C_1 / P_C_1

print(f'P_A_0_C_1={P_A_0_C_1}')
print(f'P_A_1_C_1={P_A_1_C_1}')
```

```
P_A_0_C_1=0.5
P_A_1_C_1=0.5
```

### 4.2.2 $P(B|C = 1)$

[5]:
```
P_B_0C_1 = .27+.27
P_B_1C_1 = .03+.03

P_B_0_C_1 = P_B_0C_1 / P_C_1
P_B_1_C_1 = P_B_1C_1 / P_C_1

print(f'P_B_0_C_1={P_B_0_C_1}')
print(f'P_B_1_C_1={P_B_1_C_1}')
```

```
P_B_0_C_1=0.8999999999999999
P_B_1_C_1=0.09999999999999998
```

$$\therefore P(B = 0|C = 1) = 0.9, P(B = 1|C = 1) = 0.1$$

### 4.2.3 $P(A, B|C = 1)$

[6]:
```
P_A_0B_0C_1 = .27
P_A_0_B_0_C_1 = P_A_0_C_1 * P_A_0B_0C_1 / P_A_0C_1

P_A_1B_0C_1 = .27
P_A_1_B_0_C_1 = P_A_1_C_1 * P_A_1B_0C_1 / P_A_1C_1

P_A_0B_1C_1 = .03
P_A_0_B_1_C_1 = P_A_0_C_1 * P_A_0B_1C_1 / P_A_0C_1

P_A_1B_1C_1 = .03
P_A_1_B_1_C_1 = P_A_1_C_1 * P_A_1B_1C_1 / P_A_1C_1
```

```python
print(f'P_A_0_B_0_C_1={P_A_0_B_0_C_1}')
print(f'P_A_1_B_0_C_1={P_A_1_B_0_C_1}')
print(f'P_A_0_B_1_C_1={P_A_0_B_1_C_1}')
print(f'P_A_1_B_1_C_1={P_A_1_B_1_C_1}')
```

```
P_A_0_B_0_C_1=0.44999999999999996
P_A_1_B_0_C_1=0.44999999999999996
P_A_0_B_1_C_1=0.04999999999999999
P_A_1_B_1_C_1=0.04999999999999999
```

$$\therefore \begin{aligned} P(A=0,B=0|C=1) &= 0.45 \\ P(A=1,B=0|C=1) &= 0.45 \\ P(A=0,B=1|C=1) &= 0.05 \\ P(A=1,B=1|C=1) &= 0.05 \end{aligned}$$

### 4.3 (c)

We have

$$(A \perp\!\!\!\perp B)|C \Leftrightarrow P(A,B|C) = P(A|C)P(B|C) \Leftrightarrow$$
$$P(A=i, B=j|C=k) = P(A=i|C=k)P(B=j|C=k)$$
$$\forall i \in \{0,1\}, j \in \{0,1\}, k \in \{0,1\}$$

which we can show by testing all 8 combinations:

[16]:
```
[P_A_0_B_0_C_0-P_A_0_C_0*P_B_0_C_0,
P_A_1_B_0_C_0-P_A_1_C_0*P_B_0_C_0,
P_A_0_B_1_C_0-P_A_0_C_0*P_B_1_C_0,
P_A_1_B_1_C_0-P_A_1_C_0*P_B_1_C_0,
P_A_0_B_0_C_1-P_A_0_C_1*P_B_0_C_1,
P_A_1_B_0_C_1-P_A_1_C_1*P_B_0_C_1,
P_A_0_B_1_C_1-P_A_0_C_1*P_B_1_C_1,
P_A_1_B_1_C_1-P_A_1_C_1*P_B_1_C_1]
```

[16]:
```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

The condition is satisfied for all combinations, hence $(A \perp\!\!\!\perp B)|C$.

## 4.4 (d)

[8]:
```
P_A_0 = .096+.024+.27+.03
P_A_1 = .224+.056+.27+.03
P_B_0 = .096+.224+.27+.27
P_B_1 = .024+.056+.03+.03

print(f'P_A_0={P_A_0-.00000000000000004}') # Floating point errors
print(f'P_A_1={P_A_1-.0000000000000001}')
print(f'P_B_0={P_B_0-.0000000000000001}')
print(f'P_B_1={P_B_1}')
```

```
P_A_0=0.42
P_A_1=0.58
P_B_0=0.86
P_B_1=0.14
```

[9]:
```
P_A_0_B_0 = .096+.27
P_A_1_B_0 = .224+.27
P_A_0_B_1 = .024+.03
P_A_1_B_1 = .056+.03

print(f'P_A_0_B_0={P_A_0_B_0}')
print(f'P_A_1_B_0={P_A_1_B_0}')
print(f'P_A_0_B_1={P_A_0_B_1}')
print(f'P_A_1_B_1={P_A_1_B_1}')
```

```
P_A_0_B_0=0.366
P_A_1_B_0=0.494
P_A_0_B_1=0.054
P_A_1_B_1=0.086
```

### 4.5 (e)

$A$ independent of $B \Leftrightarrow P(A=i, B=j) = P(A=i)P(B=j)\,\forall i \in \{0,1\}, j \in \{0,1\}$. We can explore this by trying all four combinations:

[18]:
```
[P_A_0_B_0 - P_A_0*P_B_0,
P_A_0_B_1 - P_A_0*P_B_1,
P_A_1_B_0 - P_A_1*P_B_0,
P_A_1_B_1 - P_A_1*P_B_1]
```

[18]:
```
[0.0047999999999999154,
 -0.004800000000000013,
 -0.0048000000000001375,
 0.004799999999999971]
```

As the probabilies are not equal, $A$ and $B$ are not independent.

# 5

## 5.1 (a)

From lecture, we have that ML-estimate of the parameters $\theta$ are given by:

$$\theta_0 = N_1/N$$

$$\theta_{j,0} = \frac{\sum_{i=1}^{N} \mathbb{1}[X_{j,i} = 1, Y_i = 0]}{\sum_{i=1}^{N} \mathbb{1}[Y_i = 0]}$$
$$\theta_{j,1} = \frac{\sum_{i=1}^{N} \mathbb{1}[X_{j,i} = 1, Y_i = 1]}{\sum_{i=1}^{N} \mathbb{1}[Y_i = 1]}$$

I will calculate these in Python:

[11]:
```
import pandas as pd
import numpy as np

table = {
    'O': [0,1,0,1,1,1,1,0],
    'B': [0,1,1,0,1,0,1,0],
    'C': [0,0,1,0,1,1,0,1],
    'A': [1,0,1,1,0,0,1,1],
    'G': [1,0,1,1,0,1,1,1]
}
table = pd.DataFrame(data=table)
```

[12]:
```
y = np.array(table['G'])
X = np.array(table[['O','B','C','A']])
k = X.shape[1]

N_1 = sum(y)
N = y.shape[0]

theta_0 = N_1 / N

thetas = np.empty([k, 2])

for i in range(k):
    thetas[i,0] = sum((X.T[i] == 1) & (y == 0)) / sum(y == 0)
    thetas[i,1] = sum((X.T[i] == 1) & (y == 1)) / sum(y == 1)

print(f'theta_0={theta_0}')
print(f'theta_i,0={thetas.T[0]}')
print(f'theta_i,1={thetas.T[1]}')
```

```
theta_0=0.75
theta_i,0=[1.  1.  0.5 0. ]
theta_i,1=[0.5        0.33333333 0.5        0.83333333]
```

### 5.2 (b)

By the assumption of conditional independence, we have

$$\begin{aligned} P(G_i)P(O_i, B_i, C_i, A_i|G_i) &= P(G_i)P(O_i|G_i)P(B_i|G_i)P(C_i|G_i)P(A_i|G_i) \\ &= \theta_0^{\mathbb{1}(G_i=1)}(1-\theta_0)^{\mathbb{1}(G_i=0)} \prod_{j=1}^{N} \theta_{j,1}^{\mathbb{1}(X_{j,i}=1)}(1-\theta_{j,0})^{\mathbb{1}(X_{j,i}=0)} \end{aligned}$$

[13]:
```
X = np.array([
    [0,1,0,1], #Example 9
    [1,1,1,1]  #Example 10
])

for e in range(X.shape[0]):
    for y in [0, 1]:
        P = y*theta_0 + (1-y)*(1-theta_0)
        for j in range(k):
            P = P * (X[e, j]*thetas[j, y] + (1-X[e, j])*(1-thetas[j, y]))
        print(f'Example #{9+e}, P(y={y}|X)={P}')
```

```
Example #9, P(y=0|X)=0.0
Example #9, P(y=1|X)=0.052083333333333336
Example #10, P(y=0|X)=0.0
Example #10, P(y=1|X)=0.052083333333333336
```

Since $P(Y=1|X) > P(Y=0|X)$ for both examples, we would classify both as good restaurants.

### 5.3 (c)

[14]:
```
y = np.array(table['G'])
X = np.array(table[['O','B','C','A']])
k = X.shape[1]

N_1 = sum(y)
N = y.shape[0]

theta_0 = N_1 / N

thetas = np.empty([k, 2])

for i in range(k):
    thetas[i,0] = (sum((X.T[i] == 1) & (y == 0)) + 1) / (sum(y == 0) + 2)
    thetas[i,1] = (sum((X.T[i] == 1) & (y == 1)) + 1) / (sum(y == 1) + 2)

print(f'theta_0={theta_0}')
print(f'theta_i,0={thetas.T[0]}')
print(f'theta_i,1={thetas.T[1]}')
```

```
theta_0=0.75
theta_i,0=[0.75 0.75 0.5  0.25]
theta_i,1=[0.5   0.375 0.5   0.75 ]
```

No more $\theta = 0$!

### 5.4 (d)

[15]:
```
X = np.array([
    [0,1,0,1], #9
    [1,1,1,1]  #10
])

for e in range(X.shape[0]):
    for y in [0, 1]:
        P = y*theta_0 + (1-y)*(1-theta_0)
        for j in range(k):
            P = P * (X[e, j]*thetas[j, y] + (1-X[e, j])*(1-thetas[j, y]))
        print(f'Example #{9+e}, P(y={y}|X)={P}')
```

```
Example #9, P(y=0|X)=0.005859375
Example #9, P(y=1|X)=0.052734375
Example #10, P(y=0|X)=0.017578125
Example #10, P(y=1|X)=0.052734375
```

Since $P(Y = 1|X) > P(Y = 0|X)$ for both examples, we would again classify both as good restaurants.

# 6

## 6.1 (a)

Using the NB assumtion we have:

$$
\begin{aligned}
&P(x^{(1)}, \cdots, x^{(m)}, y^{(1)}, \cdots, y^{(m)}) \\
&= \prod_{i=1}^{N} P(y^{(i)}) P(x^{(i)} | y^{(i)}) \\
&= \prod_{i=1}^{N} P(y^{(i)}) \prod_{j=1} \prod_{k=1}^{s} P(x_j^{(i)} = k | y^{(i)}) \\
&= \prod_{i=1}^{N} \theta_0^{\mathbb{1}[y^{(i)}=1]} (1 - \theta_0)^{\mathbb{1}[y^{(i)}=0]} \prod_{j=1} \underbrace{\theta_{j,s|y}^{\mathbb{1}[x^{(i)}=s]}}_{=1-\sum_{k=1}^{s-1} \theta_{j,k|y}} \prod_{k=1}^{s-1} \theta_{j,k|y}^{\mathbb{1}[x^{(i)}=k]}
\end{aligned}
$$

### 6.2 (b)

To simplify minimzation, we first take the log likelihood. We then minimize by finding a stationary point $\nabla P = 0$

$$
\begin{aligned}
&\log P(x^{(1)}, \cdots, x^{(m)}, y^{(1)}, \cdots, y^{(m)}) \\
&= \log \Big[ \prod_{i=1}^{N} \theta_0^{\mathbb{1}[y^{(i)}=1]} (1-\theta_0)^{\mathbb{1}[y^{(i)}=0]} \prod_{j=1} \underbrace{\theta_{j,s|y}^{\mathbb{1}[x^{(i)}=s]}}_{=1-\sum_{k=1}^{s-1}\theta_{j,k|y}} \prod_{k=1}^{s-1} \theta_{j,k|y}^{\mathbb{1}[x^{(i)}=k]} \Big] \\
&= \sum_{i=1}^{N} \left[ \mathbb{1}[y^{(i)}=1] \log \theta_0 + \mathbb{1}[y^{(i)}=0] \log(1-\theta_0) \right] + \sum_{j=1} \mathbb{1}[x^{(i)}=s] \log \theta_{j,s|y} + \sum_{k=1}^{s-1} \mathbb{1}[x^{(i)}=k] \log \theta_{j,k|y}
\end{aligned}
$$

For $\theta_0$ we have

$$
\begin{aligned}
\frac{\partial \log P}{\partial \theta_0} &= 0 && \Leftrightarrow \\
\frac{1}{\theta_0} \sum_{i=1}^{N} \mathbb{1}[y^{(i)}=1] - \frac{1}{1-\theta_0} \sum_{i=1}^{N} \mathbb{1}[y^{(i)}=0] &= 0 && \Leftrightarrow \\
\theta_0 \underbrace{\Big(\sum_{i=1}^{N} \mathbb{1}[y^{(i)}=1] + \sum_{i=1}^{N} \mathbb{1}[y^{(i)}=0]\Big)}_{=N} &= \sum_{i=1}^{N} \mathbb{1}[y^{(i)}=1] && \Leftrightarrow \\
\theta_0 &= \frac{\sum_{i=1}^{N} \mathbb{1}[y^{(i)}=1]}{N}
\end{aligned}
$$

For $\theta_{j,k|y}, k = 1, \ldots, s$ and $y_i = y$ we have

$$
\begin{aligned}
\frac{\partial \log P}{\partial \theta_{j,k|y}} &= 0 && \Leftrightarrow \\
\frac{1}{\theta_{j,k|y}} \sum_{i=1}^{N} \mathbb{1}[x_j^{(i)}=k, y^{(i)}=y] + \frac{1}{1-\theta_{j,k|y}} \sum_{i=1}^{N} \mathbb{1}[x_j^{(i)} \neq k, y^{(i)}=y] &= 0 && \Leftrightarrow \\
\theta_{j,k|y} \underbrace{\Big(\sum_{i=1}^{N} \mathbb{1}[x_j^{(i)}=k, y^{(i)}=y] + \sum_{i=1}^{N} \mathbb{1}[x_j^{(i)} \neq k, y^{(i)}=y]\Big)}_{\sum_{i=1}^{N} \mathbb{1}[y^{(i)}=y]} &= \sum_{i=1}^{N} \mathbb{1}[x_j^{(i)}=k, y^{(i)}=y] && \Leftrightarrow \\
\theta_{j,k|y} &= \frac{\sum_{i=1}^{N} \mathbb{1}[x_j^{(i)}=k, y^{(i)}=y]}{\sum_{i=1}^{N} \mathbb{1}[y^{(i)}=y]}
\end{aligned}
$$

$$\therefore$$

$$
\begin{aligned}
\theta_0 &= \frac{\sum_{i=1}^{N} \mathbb{1}[y^{(i)}=1]}{N} \\
\theta_{j,k|y=0} &= \frac{\sum_{i=1}^{N} \mathbb{1}[x_j^{(i)}=k, y^{(i)}=0]}{\sum_{i=1}^{N} \mathbb{1}[y^{(i)}=0]} \\
\theta_{j,k|y=1} &= \frac{\sum_{i=1}^{N} \mathbb{1}[x_j^{(i)}=k, y^{(i)}=1]}{\sum_{i=1}^{N} \mathbb{1}[y^{(i)}=1]}
\end{aligned}
$$

The expression holds for all $k$, including $k = s$

This means that the probability of a feature $j$ having $x_j^{(i)} = k$ for a given $k$ is proportional to the fraction of observations where $x_j^{(i)} = k$. This is weighted by the relative frequency of such observations with $y^{(i)} = y$ and observatons with $y^{(i)} = y$ in general in the data set.