COMP 6321 Machine Learning Assignment

Qingbo Kang, Concordia University 40058122

1 A midterm preparation question

1.1 a

For this problem, I think instance-based learning algorithm such as k-NN may be the best learning algorithm to use. Because the training set is larger than the queries, so the training time must taken into account. The training time for k-NN is $O(n)$. k-NN just store the training set in training phase. An appropriate value of k can be found via cross-validation.

1.2 b

For this problem, I think decision tree algorithm may be the best learning algorithm to use. Because the classifier has to be justified to the board of education before it is implemented, and one of the advantages of decision tree is simple to understand and interpret, people are able to understand decision tree models after a brief explanation.

1.3 c

For this problem, I think Naive Bayes classifier may be the best model to use. This problem have binary input and binary output which is very easy to fit in. The features in this problem seems independence and this is the core assumption of Naive Bayes. In addition, Naive Bayes is actually a probability table that gets updated through the training data which is more suitable for news data comes in, the classifier will have to be updated frequently and efficiently.

1.4 d

For this problem, I would suggest SVM may be the best learning algorithm to use. SVM can model non-linear decision boundaries. It's also very robust against overfitting especially in high-dimensional feature space.

2 Properties of entropy

2.1 a

According to the given joint probabilities, we can compute the probabilities for X and Y:

$$\begin{aligned} p(x=0) &= p(0,0)+p(0,1) = \frac{2}{3} \\ p(x=1) &= p(1,0)+p(1,1) = \frac{1}{3} \\ p(y=0) &= p(0,0)+p(1,0) = \frac{1}{3} \\ p(y=1) &= p(0,1)+p(1,1) = \frac{2}{3} \end{aligned}$$

2.1.1 i

$$H[x] = -[p(x=0)\log_2 p(x=0) + p(x=1)\log_2 p(x=1)] = -[\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}] = 0.9183$$

2.1.2 ii

$$H[y] = -[p(y=0)\log_2 p(y=0) + p(y=1)\log_2 p(y=1)] = -[\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}] = 0.9183$$

2.1.3 iii

$$\begin{aligned} H[y|x] &= \sum_v p(x=v)H(y|x=v) \\ &= p(x=0)H(y|x=0) + p(x=1)H(y|x=1) \\ &= \frac{2}{3}\log_2\frac{\frac{2}{3}}{\frac{1}{3}} + \frac{1}{3}\log_2\frac{\frac{1}{3}}{\frac{1}{3}} \\ &= \frac{2}{3} + \frac{1}{3}\times 0 \\ &= \frac{2}{3} \end{aligned}$$

2.1.4 iv

$$\begin{aligned} H[x|y] &= \sum_v p(y=v)H(x|y=v) \\ &= p(y=0)H(x|y=0) + p(y=1)H(x|y=1) \\ &= \frac{1}{3}\log_2\frac{\frac{1}{3}}{\frac{1}{3}} + \frac{2}{3}\log_2\frac{\frac{2}{3}}{\frac{1}{3}} \\ &= 0 + \frac{2}{3}\times 1 \\ &= \frac{2}{3} \end{aligned}$$

2.1.5 v

$$H[x,y] = H[y|x] + H[x] = \frac{2}{3} + 0.9183 = 1.5850$$

2.1.6 vi

$$\begin{aligned} I[x,y] &= \sum_x\sum_y p(x,y)\log_2(\frac{p(x,y)}{p(x)p(y)}) \\ &= \frac{1}{3}\log_2\frac{\frac{1}{3}}{\frac{2}{3}\times\frac{1}{3}} + \frac{1}{3}\log_2\frac{\frac{1}{3}}{\frac{2}{3}\times\frac{2}{3}} \\ &= \frac{2}{3}\log_2\frac{3}{2} + \frac{1}{3}\log_2\frac{3}{4} \\ &= 0.2516 \end{aligned}$$

2.2 b

We wish to find:

$$\arg\max_{p_n}\sum_{n=1}^{N} p_n\log(p_n)$$

with constraints:

$$\sum_{n=1}^{N} p_n = 1, p_i \geq 0, i = 1, 2, ..., N$$

Using Lagrange for maximization with constraints with a Lagrangian multiplier only for the first constraint:

$$L(p_1, p_2, ..., p_n, \lambda) = \sum_{n=1}^{N} p_n \log(p_n) - \lambda(1 - \sum_{n=1}^{N} p_n)$$

Setting the gradient of the Lagrangian function to 0, that is:

$$\frac{\partial L}{\partial p_1} \sum_{n=1}^{N} p_n \log(p_n) - \lambda(1 - \sum_{n=1}^{N} p_n) = 0$$
$$\frac{\partial L}{\partial p_2} \sum_{n=1}^{N} p_n \log(p_n) - \lambda(1 - \sum_{n=1}^{N} p_n) = 0$$
$$...$$
$$\frac{\partial L}{\partial p_N} \sum_{n=1}^{N} p_n \log(p_n) - \lambda(1 - \sum_{n=1}^{N} p_n) = 0$$
$$\frac{\partial L}{\partial \lambda} \sum_{n=1}^{N} p_n \log(p_n) - \lambda(1 - \sum_{n=1}^{N} p_n) = 0$$

and we yield

$$\log(p_1) + 1 - \lambda p_1 = 0$$
$$\log(p_2) + 1 - \lambda p_2 = 0$$
$$...$$
$$\log(p_N) + 1 - \lambda p_N = 0$$
$$\sum_{n=1}^{N} p_n = 1$$

From which:

$$\lambda = \frac{\log(p_1) + 1}{p_1} = \frac{\log(p_2) + 1}{p_2} = ... = \frac{\log(p_N) + 1}{p_N}$$

From these above equations, it is clear that $p_1 = p_2 = ... = p_N = \frac{1}{N}$, which is an uniform distribution.

2.3 c

The constraints can be written as:

$$\int_{-\infty}^{\infty} f_j(x)p(x)dx = a_j$$

We consider the functional:

$$J(p(x)) = \int_{-\infty}^{\infty} p(x)\ln p(x)dx - \lambda_0(\int_{-\infty}^{\infty} p(x)dx - 1) - \sum_{j=1}^{n} \lambda_j(\int_{-\infty}^{\infty} f_j(x)p(x)dx - a_j)$$

where λ_j are the Lagrange multipliers. The zeroth constraint ensures the second axiom of probability. The other constraints are that the measurements of the function are given constants up to order n. The entropy attains an extremum when the functional derivative is equal to 0:

$$\frac{\delta J(p(x))}{\delta p(x)} = \ln p(x) + 1 - \lambda_0 - \sum_{j=1}^{n} \lambda_j f_j(x) = 0$$

This extremum is a maximum. Therefore, the maximum entropy probability distribution in this case is in this form:

$$p(x) = e^{-1+\lambda_0} e^{\sum_{j=1}^{n} \lambda_j f_j(x)} = c\exp(\sum_{j=1}^{n} \lambda_j f_j(x))$$

which is a Gaussian distribution.

2.4 d

In order to prove test T1 wins, we have to prove that test T1 yield the maximum mutual information or information gain, i.e. $IG(D, T1) > IG(D, T2)$. Firstly, we compute $H(D)$:

$$H(D) = -\frac{3}{4} log_2 \frac{3}{4} - \frac{1}{4} log_2 \frac{1}{4} = 0.8113$$

Then compute the information gain of test T1:

$$H(D|T1) = \frac{3}{4}(-\frac{2}{3}\log_2 \frac{2}{3} - \frac{1}{3}\log_2 \frac{1}{3}) + \frac{1}{4} \times 0 = 0.6887$$
$$IG(D, T1) = H(D) - H(D|T1) = 0.8113 - 0.6887 = 0.1226$$

On the other hand, the information gain of test T2:

$$H(D|T2) = \frac{22}{40}(-\frac{15}{22}\log_2 \frac{15}{22} - \frac{7}{22}\log_2 \frac{7}{22}) + \frac{18}{40}(-\frac{15}{18}\log_2 \frac{15}{18} - \frac{3}{18}\log_2 \frac{3}{18}) = 0.7888$$
$$IG(D, T2) = H(D) - H(D|T2) = 0.8113 - 0.7888 = 0.0225$$

It can be clearly seen that the information gain of test T1 is bigger than the information gain of test T2. i.e. $IG(D, T1) > IG(D, T2)$. Therefore, the test T1 wins.

3 Kernels

3.1 a

Since K_1, K_2 are kernels, $K_{1ij} = K_{1ji}, K_{2ij} = K_{2ji}$. Both kernels matrices are positive semidefinite.

$$K_{ij} = aK_{1ij} + bK_{2ij} = aK_{1ji} + bK_{2ji} = K_{ji}$$

it is symmetric.

$$xKx^T = x(aK_1 + bK_2)x^T = xaK_1x^T + xbK_2x^T$$

since $a, b > 0$, K is positive semidefinite. Thus, by Merceris Theorem, this function is a kernel.

3.2 b

Suppose:

$$a = 1, b = 1, K_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, K_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Both K_1 and K_2 are symmetric, positive and semidefinite matrices, it would result in:

$$K(x, z) = aK_1 - bK_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The eigenvalues of which are $\lambda_1 = -1, \lambda_2 = 1$, making $K(x, z)$ a non-positive semidefinite matrix and thus $K(x, z)$ is not a valid kernel.

3.3 c

Let $K_1(x, z) = \phi_1(x)\phi_1(z)$ and $K_2(x, z) = \phi_2(x)\phi_2(z)$. Then, use the definition of kernel:

$$\begin{aligned} K(x, z) &= (\phi_1(x)^T\phi_1(z))(\phi_2(x)^T\phi_2(z)) \\ &= \sum_{i=1}^{a} \phi_1(x)_i\phi_1(z)_i \times \sum_{j=1}^{b} \phi_2(x)_j\phi_2(z)_j \\ &= \sum_{i,j=1}^{a,b} (\phi_1(x)_i\phi_2(x)_j)(\phi_1(z)_i\phi_2(z)_j) \end{aligned}$$

since $\phi_1(x)$ and $\phi_2(x)$ are given functions. Thus, $K(x, z)$ is a kernel.

3.4 d

Using the definition of kernel function:

$$K(x, z) = \phi(x)\phi(z)$$

Then we let $\phi_1(x) : \mathbb{R}^n \to \mathbb{R}$, then $\phi(x)^T = \phi(x)$. The original formula can also be written in the form of a kernel. Thus. $K(x, z)$ is a kernel.

3.5 e

The same rational as question 3.d applies here. $K(x, z) = p(x)p(z)$ is a kernel.

4 Nearest neighbor vs decision trees

For these two classification approaches, in general or most cases, the boundaries are not coincide. In fact, they tend to be non-coincidental but in some rare cases, the boundaries might be equal to. boundaries for NN form a Voronoi tessellation, where each boundary segment corresponds to a hyper-plan running orthogonal to the line between a given point and its nearest neighbors while passing through the midpoint of such a line. On the other hand, decision tree boundaries are composed of hyper-planes that are orthogonal to the features f_d chosen for each decision, boundaries pass through the midpoint between points neighboring on a projection along the axis of f_d. Thus each segment of a decision tree boundary will generally have one out of n directions for an n-dimensional space.

5

5.1 a

The minimal multi-class classification error rate P^* is given by:

$$P^* = 1 - \int \max_i P(\omega_i|x)P(x)dx$$

since that $P(\omega_i|x)P(x) = P(x, \omega_i) = P(x|\omega_i)P(\omega_i)$, thus:

$$P(\omega_i|x)P(x) = \begin{cases} \frac{1}{c} & 0 \leq x \leq \frac{cr}{c-1} \\ \frac{1}{c} & i \leq x \leq i + 1 - \frac{cr}{c-1} \\ 0 & \text{elsewhere} \end{cases}$$

Given the class density and probability, we can see that for any region with overlapping densities, the choice of any i will maximize, in addition, it can be seen that the constraints imposed by existing densities demand that $0 \leq r \leq \frac{c-1}{c}$. This in turn implies that densities overlap only in $[0, \frac{cr}{c-1}]$ thus:

$$\begin{aligned} P^* &= 1 - \int P(\omega_1|x)P(x)dx \\ &= 1 - \frac{1}{c}\int_0^{\frac{cr}{c-1}} 1dx - \sum_{i=1}^{c} \frac{1}{c}\int_i^{i+1-\frac{cr}{c-1}} 1dx \\ &= 1 - \frac{1}{c}\frac{cr}{c-1} - 1 - \frac{cr}{c-1} \\ &= \frac{cr - r}{c - 1} \\ &= r \end{aligned}$$

5.2 b

From the piece-wise densities, the class prior and the fact that $p(x) = \sum_{i=1}^{c} P(x|\omega_i)P(\omega_i)$, we can derive:

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{p(x)} = \begin{cases} \frac{1}{c} & 0 \leq x \leq \frac{cr}{c-1} \\ 1 & i \leq x \leq i+1-\frac{cr}{c-1} \\ 0 & \text{elsewhere} \end{cases}$$

Thus:

$$\begin{aligned} L_{NN} &= \int [1 - \sum_{i=1}^{c} P^2(\omega_i|x)]p(x)dx \\ &= \int [1 - \sum_{i=1}^{c} (\frac{P(x|\omega_i)P(\omega_i)}{p(x)})^2]p(x)dx \\ &= \int p(x) - \sum_{i=1}^{c} \frac{P(x|\omega_i)^2 P(\omega_i)^2}{p(x)} dx \\ &= \int p(x) - \sum_{i=1}^{c} \frac{P(x|\omega_i)P(\omega_i)(P(\omega_i|x)p(x))}{p(x)} dx \\ &= \int p(x) - \sum_{i=1}^{c} P(x|\omega_i)P(\omega_i)p(\omega_i|x)dx \\ &= 1 - \frac{1}{c}\sum_{i=1}^{c} \int_0^{\frac{cr}{c-1}} \frac{1}{c}dx - \sum_{i=1}^{c} \frac{1}{c} \int_i^{i+1-\frac{cr}{c-1}} 1dx \\ &= 1 - \frac{1}{c}\frac{cr}{c-1} - 1 - \frac{cr}{c-1} \\ &= \frac{cr-r}{c-1} \\ &= r \end{aligned}$$

Therefore, for this case that the nearest-neighbor rate is $P = P^*$.

6 Implementation

I choose to perform (b). Code can be seen in Q6b.m and knn.m.
Figure 1 shows the training and testing error curves as a function of k.

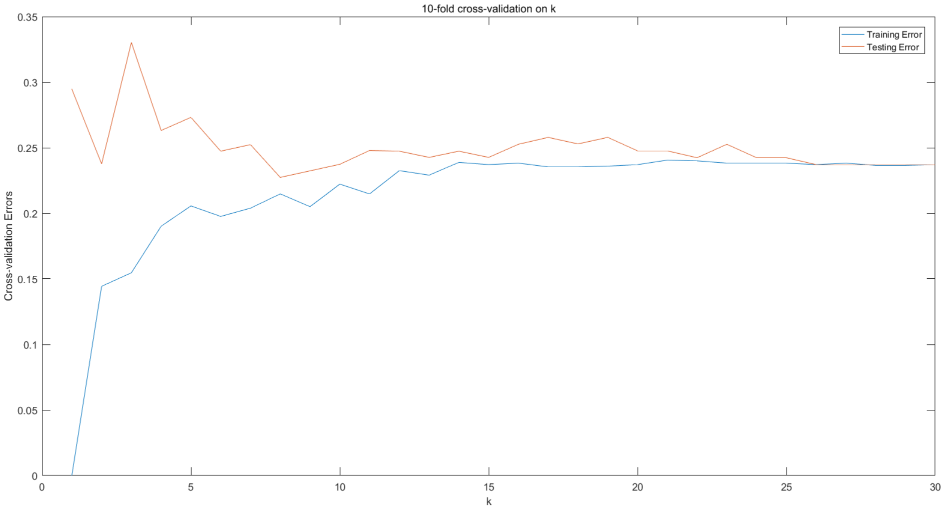

Figure 1: Training and testing error curves