Assignment 1

CSE5002 Intelligent Data Analysis (Spring 2022)

Name: Jiang Yuchen ID: 11812419

Q1

(1) We need regularization to avoid overfitting and increase the robustness when training models.

(2) L1-norm is $||w||_1 = \sum_{i=1}^{N} |w_i|$ and L2-norm is $||w||_2 = (\sum_{i=1}^{N} |w_i|^2)^{\frac{1}{2}}$.

We often use L1-norm to get a sparse model since its derivative is a constant and it will turn the weak or irrelevant features into zero. L2-norm is convenient and fast to compute and the solutions are more average than L1-norm due to square penalty.

(3) According to (2), L1-norm will be used to exclude those irrelevant and redundant features since it will reduce those features into zero. The objective function is

$Loss(w) = Error(w) + R(w) = (y - Xw)^T(y - Xw) + \lambda||w||_1^2$ and we need to minimize it.

Q2

(1) w is perpendicular to decision boundary.

(2) $Margin\ distance\ = ||x^+ - x^-||_2 = ||\lambda w||_2 = \frac{2}{||w||_2}$.

(3) It depends. If the additional features are noisy, then $||w_{new}||$ will be greater to distinguish positive and negative labels. And if the additional features are helpful, $||w_{new}||$ will be smaller since it's easy to divide two kinds of labels and even small weights can separate them into positive and negative results. Otherwise, it will be same.

(4) From space complexity, since kernel trick will increase the dimension of the original data, we need more space to store the information of additional dimension. Thus, the space complexity will change.

From time complexity, it's obvious that kernel trick will calculate kernel function for non-linear data instead of directly multiplying w towards data. Thus, the time complexity will change.

Q3

For nonlinear separable data in Logistic Regression, we need to turn all x_i into $\phi(x_i)$ in $\hat{y} = 1/(1 + exp(-\langle w, x_i\rangle - b))$.

Thus, $\hat{y} = 1/(1 + exp(-\langle w, \phi(x_i)\rangle - b))$.

According to Representer Theorem, $w = \sum_{i=1}^{n} \alpha_i y_i x_i$, so $w = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i)$. (Here, we consider b as w_0)

Then, $\hat{y} = 1/(1 + exp(-\langle w, \phi(x_i)\rangle - b)) = 1/(1 + exp(-\sum_{i=1}^{n} \alpha_i y_i \phi(x_i)^T \phi(x_i)))$. Apply kernel trick, we can get $\hat{y} = 1/(1 + exp(-\sum_{i=1}^{n} \alpha_i y_i k(x_i, x)))$, which is Kernel Logistic Regression.

The objective function is $L(\hat{y}, y) = -(ylog(\hat{y}) + (1 - y)log(1 - \hat{y}))$

Suppose $z = -\sum_{i=1}^{n} \alpha_i y_i k(x_i, x)$, $\mathrm{d}\hat{y} = \frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$ and $\mathrm{d}z = \frac{\partial L}{\partial z} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} = (-\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}) \cdot \hat{y}(1 - \hat{y}) = \hat{y} - y$.

Then, $\mathrm{d}\alpha_i = \frac{\partial L}{\partial \alpha_i} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial \alpha_i} = \mathrm{d}z \cdot (-y_i k(x_i, x)) = (y - \hat{y}) y_i k(x_i, x)$

Consequently, the Kernel Logistic Regression is $\hat{y} = 1/(1 + exp(-\sum_{i=1}^{n} \alpha_i y_i k(x_i, x)))$. Objective function is $L(\hat{y}, y) = -(ylog(\hat{y}) + (1 - y)log(1 - \hat{y}))$ and the derive is $\mathrm{d}\alpha_i = (y - \hat{y}) y_i k(x_i, x)$

Q4

(1) Perceptron Unit: $y = \begin{cases} 0, if \sum_{i=0}^{n} w_i x_i < 0 \\ 1, if \sum_{i=0}^{n} w_i x_i \geq 0 \end{cases}$

Sigmoid Unit: $y = \frac{1}{1+exp(-net)}$ and $net = \sum_{i=0}^{n} w_i x_i$

(2) Universal Approximation Theorem shows that it's possible to fit any complex function when giving correct and enough parameters in simple neural network. It's the base for neural network.

(3) Learning perceptron is to update weights according to training data. First, we start with some initial values for the weights. The n, use the perceptron to classify training examples and modify weights when errors occurs.

Update formulas: $w_t = w_{t-1} + \eta(y_t - \hat{y_t})x_t$ and $\hat{y_t} = f_t(x_t) = step(\langle w_t, x_t \rangle)$

Q5

(1) According to the question, $Y : [4+, 4-]$

$X_1\ splits\ Y\ into\ Y_{10} : [1+, 1-], Y_{11} : [1+, 1-], Y_{12} : [2+, 2-]$

$X_2\ splits\ Y\ into\ Y_{20} : [2+, 0-], Y_{21} : [2+, 0-], Y_{22} : [0+, 4-]$

$X_3\ splits\ Y\ into\ Y_{31} : [1+, 0-], Y_{32} : [1+, 0-], Y_{33} : [1+, 0-], Y_{34} : [1+, 0-], Y_{35} : [0+, 1-], Y_{36} : [0+, 1-], Y_{37} : [0+, 1-], Y_{38} : [0+, 1-]$

$Entropy(Y) = -0.5log_20.5 - 0.5log_20.5 = 1$

$Gain(Y, X_1) = Entropy(Y) - 0.25Entropy(Y_{10}) - 0.25Entropy(Y_{11}) - 0.5Entropy(Y_{12}) = 1 - 0.25 - 0.25 - 0.5 = 0$

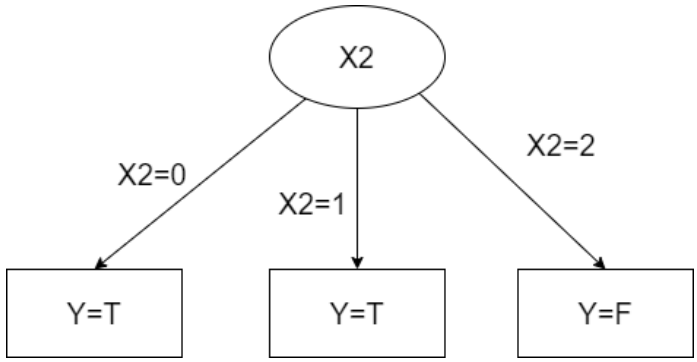
$Gain(Y, X_2) = Entropy(Y) - 0.25Entropy(Y_{20}) - 0.25Entropy(Y_{21}) - 0.5Entropy(Y_{22}) = 1 - 0 - 0 - 0 = 1$

$Gain(Y, X_3) = Entropy(Y) - 0.125Entropy(Y_{31}) - 0.125Entropy(Y_{32}) - 0.125Entropy(Y_{33}) - 0.125Entropy(Y_{33}) - 0.125Entropy($

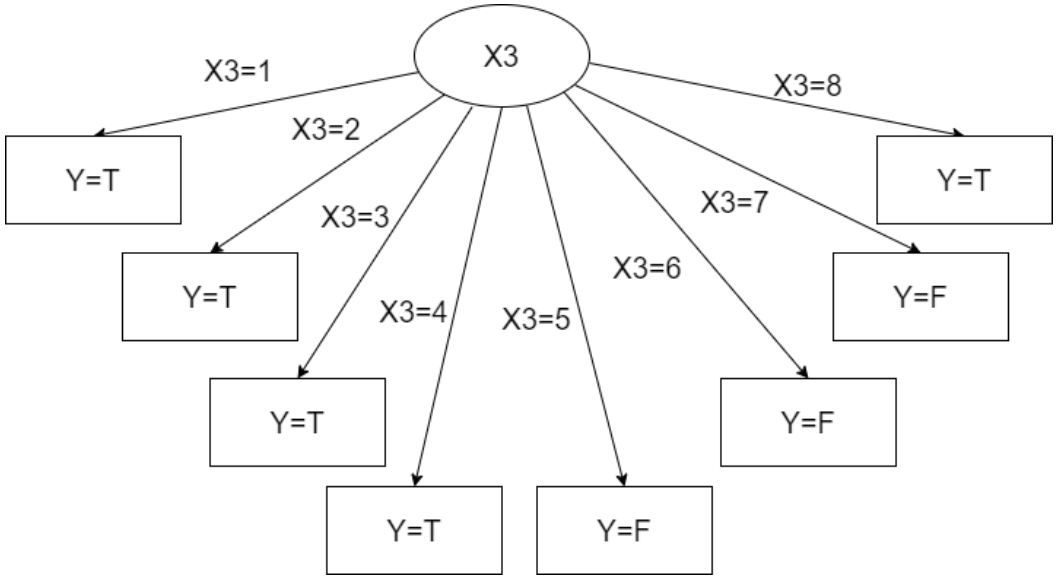
$-0.125Entropy(Y_{35}) - 0.125Entropy(Y_{36}) - 0.125Entropy(Y_{37}) - 0.125Entropy(Y_{38}) = 1$

Thus, according to the maximum information gain principle, X_2 or X_3 should be used to select in the first step of splitting the dataset.

(2) If we select X_2 ,then decision tree will be :



If we select X_3 , then decision tree will be:



(3) We can turns continuous values into discrete values, by constructing branch according to continuous values. For example, we can divide them into two groups by hand and the limit can be chosen from all possible $\frac{a_i + a_{i+1}}{2}$, $a_i\ is\ continuous\ values$. And the chosen method can also be maximum information gain principle. Thus, the final branch in decision tree will be $a_i \leq limit$ and $a_i > limit$. If needed, more groups are also possible.