CS303A Homework 3

11812419 JIANG Yuchen

Q1.

According to the question, H(y) = $-\frac{3}{5} * \log_2 \frac{3}{5} - \frac{2}{5} * \log_2 \frac{2}{5}$ = 0.9710

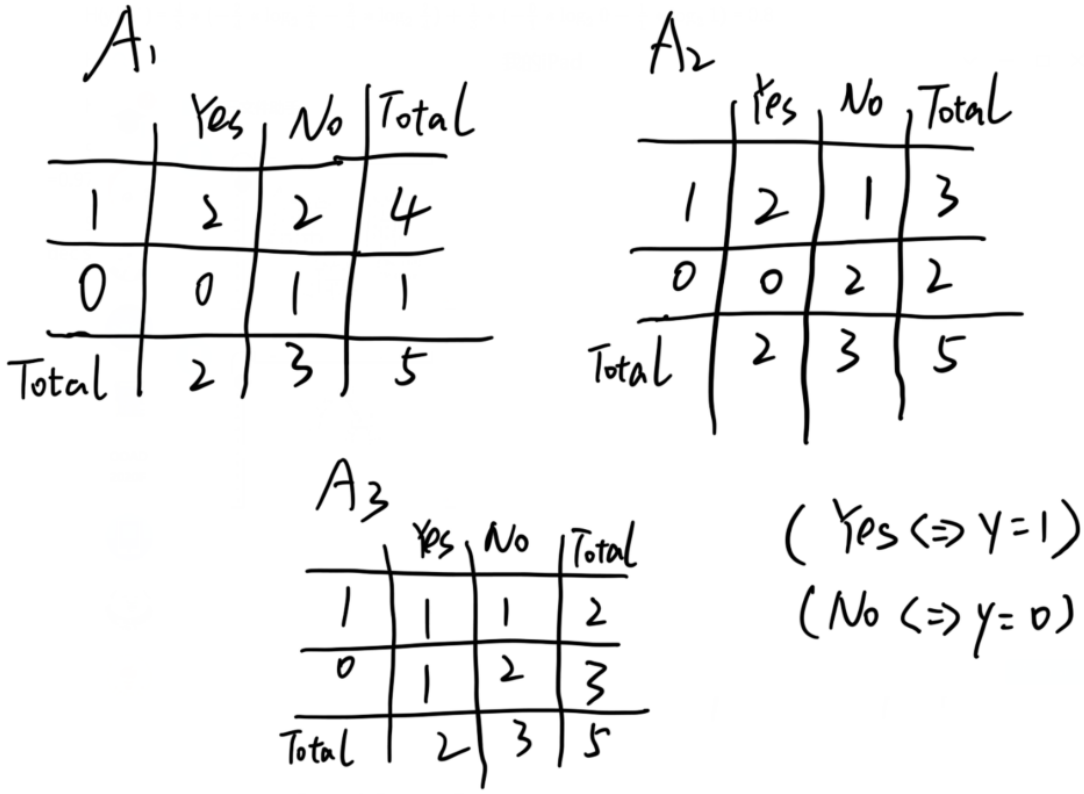

Figure.1 Entropy tables

According to the tables shown above, we can easily count entropy for each attribute.

H(y|A1) = $\frac{4}{5} * (-\frac{2}{4} * \log_2 \frac{2}{4} - \frac{2}{4} * \log_2 \frac{2}{4}) + \frac{1}{5} * (-\frac{0}{1} * \log_2 0 - \frac{1}{1} * \log_2 1)$ = 0.8

H(y|A2) = $\frac{3}{5} * (-\frac{2}{3} * \log_2 \frac{2}{3} - \frac{1}{3} * \log_2 \frac{1}{3}) + \frac{2}{5} * (-\frac{0}{2} * \log_2 0 - \frac{2}{2} * \log_2 1)$ = 0.5510

H(y|A3) = $\frac{2}{5} * (-\frac{1}{2} * \log_2 \frac{1}{2} - \frac{1}{2} * \log_2 \frac{1}{2}) + \frac{3}{5} * (-\frac{1}{3} * \log_2 \frac{1}{3} - \frac{2}{3} * \log_2 \frac{2}{3})$ = 0.9510

So Importance(A1) = H(y) - H(y|A1) = 0.9710 - 0.8 = 0.1710, Importance(A2) = H(y) - H(y|A2) = 0.9710 - 0.5510 = 0.42, Importance(A3) = H(y) - H(y|A3) =0.9710 - 0.9510 = 0.02.

Since Importance(A2) is the biggest, so we choose A2 as the root test. Then A1 and the last is A3, which are tested when constructing sub tree. The decision tree is shown below.

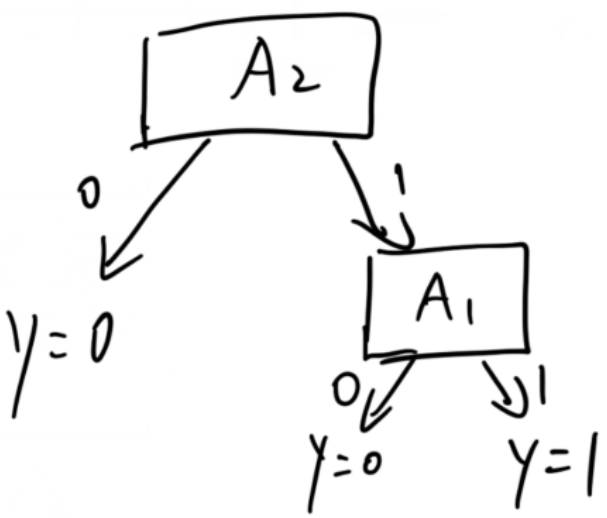

Figure.2 Decision Tree

Q2.

a.

According to the question, we can map the input [x1, x2] into [x1, x1x2].

The four inputs are [-1, -1],[-1, +1], [+1, -1], [+1, +1]. After mapping, the input become [-1, 1], [-1, -1], [+1, -1], [+1, +1]. Their output are +1, -1, -1, +1. We can find that x1x2 is related to output(x1x2 has the same value as output). Thus, the maximum margin separator is the line x1x2 = 0 with the margin = 1.

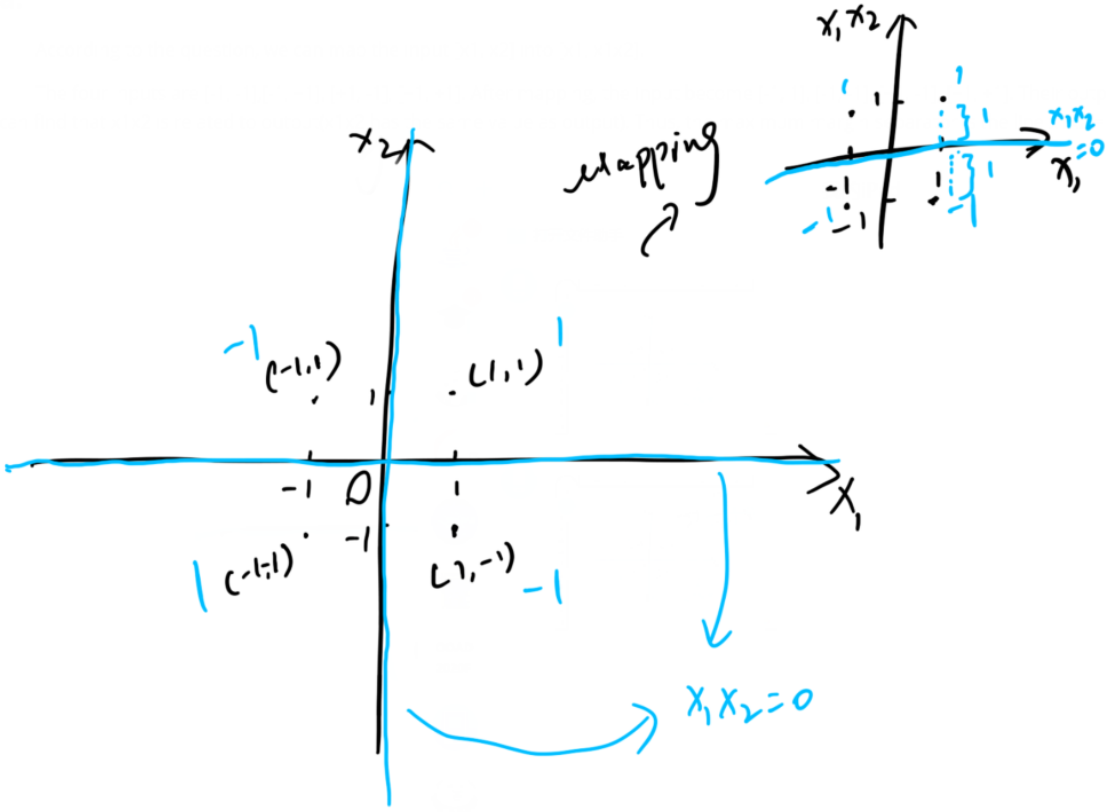

Figure.3 Mapping and Margin

b.

According to figure 3 and figure 4, the module before mapping shows the original Euclidean input space.

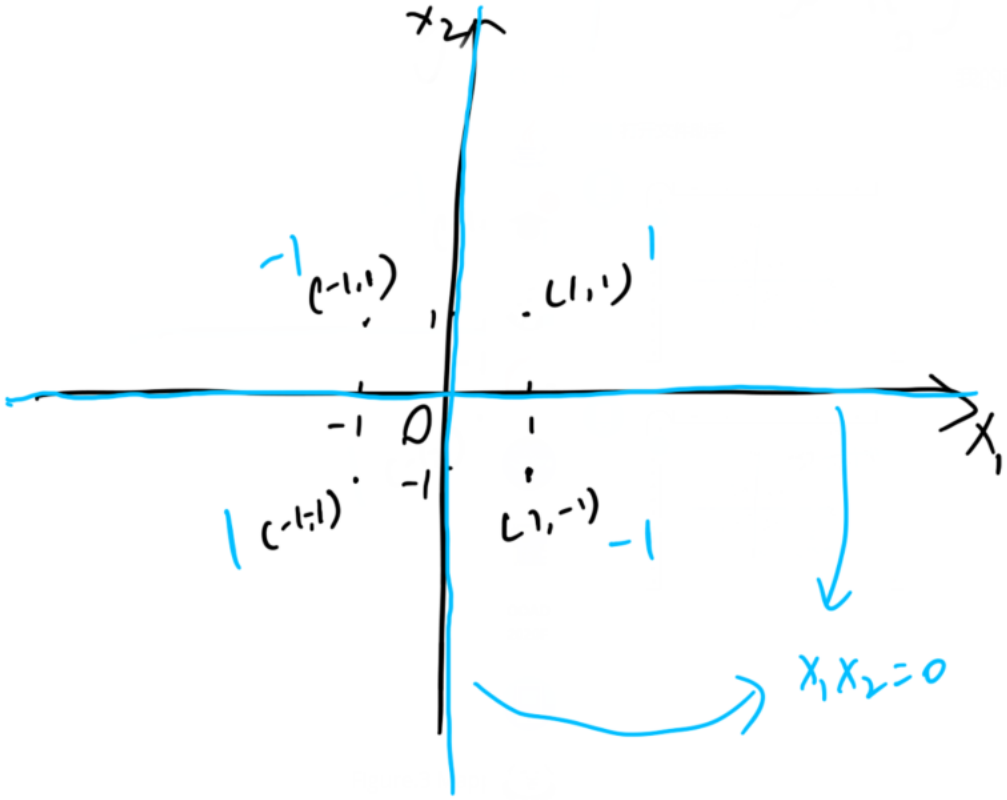

Figure.4 Module before mapping

Q3.

a.

According to the question, the output can be calculated as

$O_i = g(O_{hidden}) = g(\sum_j w_{j,i} * H_j) = g(\sum_j w_{j,i} * g(\sum_k w_{k,j} * I_k)) = c(\sum_j w_{j,i} * (c * (\sum_k w_{k,j} * I_k) + d)) + d = c^2 * \sum_k I_k \sum_j w_{k,j} w_{j,i} + d * (1 + c \sum$

So, we can transform two layers with one hidden layer into one layer whose weight is $\sum_j w_{k,j} w_{j,i}$ and activation function is that g(x) = $c^2 x + d(1 + c \sum_j w_{j,i})$, taking $\sum_k I_k$ as x.

Thus, there is a network with no hidden units that computes the same function.

b.

Similarly as a., for any number of hidden layers, what we need to do is iterating by given activation function. For a unique activation function, we can easily represent output in form of input of its layer. In such a situation, any number of hidden layers can be transformed into the form of no hidden layer.

c.

Since when we transform n input into h nodes in hidden layer we need $n * h$ matrix, so there are $2 * h$*n weights. When we reduce all hidden layer, there are n^2 weights since there are only n inputs and outputs. When h<<n, $2 * h$* ins much smaller than n^2 . Thus, reduce operation makes the network has more weight, which burden the network when processing.