ΔPROBLEM 3 [20 points]

Given an arbitrary decision tree, it might have repeated queries (splits) on some paths root-leaf. Prove that there exists an equivalent decision tree only with distinct splits on each path.

Suppose that we have a path p from root R to leaf L with repeated splits $S_i = S_j (i < j)$. S_i is between node N_x and node N_{x+1}, S_j connects N_y and N_{y+1}.(x+1 <= y)
According to the definition of decision tree, N_{x+1} is a subset of N_x, where contains all the elements satisfying S_i in N_t. Also because x+1 <= y, N_y should be a subset of N_x, which means all the elements in N_y should satisfying S_i. If we apply the split $S_j = S_i$, nothing will change, and $N_y = N_{y+1}$. That is to say, if we delete S_j, the classification will be same.
Then we can delete the entire repeat splits to get an equivalent decision tree only with distinct splits on each path.

a) Prove that for any arbitrary tree, with possible unequal branching ratios throughout, there exists a binary tree that implements the same classification functionality.

Suppose arbitrary tree T has L leaf nodes and K non-leaf nodes. N is arbitrary non-leaf nodes.
N has m children (m >= 2).

When m > 2, we can always keep the one of the m children C_i (0<i<m) and combine other children into a new node C', let C' and C_i be the new children of N. And keep C_j(0<i<m, j!= i) as one children of C', and combine all the children expect C_i and C_j into C'', let C'' and C_j to be the children of C'. We can continue this process until the new subtree S' with the root N and m leaf nodes. And in this way, we transfer the node with more than 2 children into a binary with the same classification.

And we can apply the transformation to all the nodes in the tree. Than we get a binary tree T' with the same classification as the old one.

b) Consider a tree with just two levels - a root node connected to B leaf nodes (B>=2). What are then upper and the lower limits on the number of levels in a functionally equivalent binary tree, as a function of B?

Due to the tree have only two levels and B leaf nodes, the root which contains all the data has been split into B subsets. And no matter how many level a tree has, it will be somehow been split into B subsets. As a result the best case is a complete binary tree with B leaves, the level is $\lceil log_2 B \rceil$. And the worst case is B – 1, where the deepest level contains two of the subsets and other contains only one subset. So the upper limits is B – 1, the lower limits is $\lceil log_2 B \rceil$.

c) As in b), what are the upper and lower limits on number of nodes in a functionally equivalent binary tree?

The upper limits = lower limits = 2B – 1, due to the feature of binary tree. No matter how many levels one binary tree has, if it has B leaves, it should have 2B – 1 nodes totally.

PROBLEM 4 [20 points]

Consider training a binary decision tree using entropy splits.

a) Prove that the decrease in entropy by a split on a binary yes/no feature can never be greater than 1 bit.

$\Delta\, i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$

$i(N) = -\sum_j P(w_j) log_2 P(w_j)$

And this split is a yes/no feature, which means N can only split the node N into N_a and N_b. The best case is N_L and N_R is the final classification, which means $i(N_R) = i(N_L) = 0$

$$\Delta\, i(N) = -P_L log_2 P_L - (1 - P_L) log_2 P_R - P_L i(N_L) - (1 - P_L) i(N_R) \quad (1)$$

$$\Delta\, i(N) = -P_L\left(log_2 P_L + i(N_L)\right) - (1 - P_L)(log_2 P_R + i(N_R)) \quad (2)$$

$\Delta\, i(N) = -(P_L log_2 P_L + (1 - P_L) log_2 (1 - P_L))$

When $P_L = 1/2$, $\Delta\, i(N)$ *can be the maxnum value* $= 1$

So it can never be greater than 1 bit.

b) Generalize this result to the case of arbitrary branching B>1.

When a split will split N into more than 2 subsets, N_a, N_b... can never be the final classification (leaf nodes) which means $i(N_R)$ *and* $i(N_L)$ will be less than zero. And according to (2), $\Delta\, i(N)$ will above 1

PROBLEM 5 [20 points]

Write down explicit formulas for normal equations solution presented in class for the case of one input dimension.

(Essentially assume the data is (x_i, y_i) i=1,2,..., m and you are looking for h(x) = ax+b that realizes the minimum mean square error. The problem asks you to write down explicit formulas for a and b.)

Suppose the square error is J(θ), where $\theta = (a, b)^T$

$J(\theta) = \frac{1}{2}\sum_{i=1}^{i=m}(ax_i + b - y_i)^2$

So the minimum of J(θ) happens when $\nabla J(\theta) = 0$

$\nabla J(\theta) = \nabla \frac{1}{2}\sum_{i=1}^{i=m}(ax_i + b - y_i)^2 = \nabla \frac{1}{2}\{X(a, b)^T - Y\}^T \{X(a, b)^T - Y\} = 0$

$\frac{1}{2}\nabla\{(a, b)X^T - Y^T\}\{X(a, b)^T - Y\} = \frac{1}{2}\nabla\{(a, b)X^T X(a, b)^T - (a, b)X^T Y -$

$Y^T X(\mathrm{a,b})^T + Y^T Y\}$=0 $\frac{1}{2}\nabla \mathrm{tr}\{(\mathrm{a,b})\mathrm{X}^T X(\mathrm{a,b})^T - (\mathrm{a,b})\mathrm{X}^T Y - Y^T X(\mathrm{a,b})^T + Y^T Y\}$ =

$\frac{1}{2}\nabla\{\mathrm{tr}(\mathrm{a,b})\mathrm{X}^T X(\mathrm{a,b})^T - 2tr(\mathrm{a,b})\mathrm{X}^T Y = X^T X(a\ b)^T - X^T Y = 0$

⇨ $(\mathrm{a\ b})^T = (X^T X)^{-1} X^T Y$