Week 1 - Decision Tree

0.Entropy

> **Def** measurement of random variable’s uncertainty

> **Equ** empirical entropy H(X), empirical conditional entropy H(Y|X)
> $H(Y|X) = H(X,Y) - H(X)$
> $= -\sum_{x,y} P(X,Y)logP(X,Y) + \sum_x P(X)logP(X)$
> $= -\sum_{x,y} P(X,Y)logP(X,Y) + \sum_x (\sum_y P(X,Y))logP(X)$
> $= -\sum_{x,y} P(X,Y)logP(X,Y + \sum_{x,y} P(X,Y)logP(X)$
> $= -\sum_{x,y} log\frac{P(X,Y)}{P(X)}$
> $= -\sum_{x,y} logP(Y|X)$
> $= -\sum_x \sum_y P(X)P(Y|X)logP(Y|X)$
>
> > $= -\sum_x P(X) \sum_y P(Y|X)logP(Y|X)$
>
> $= \sum_x P(X)H(Y|X = x_i)$

> **Ann** binary logarithm

1. Feature Selection

1.1 information gain

> **Def** mutual information of data set and feature

> **Equ** feature A in $(a_1, a_2, ..., a_n)$, data set D, information gain g, empirical entropy H, class number K
> $g(D, A) = H(D) - H(D|A)$
> $= -\sum_{k=1}^K P(C_k)logP(C_k) + \sum_{i=1}^n P(A_i) \sum_{k=1}^K P(D_k|A_i)logP(D_k|A_i)$
> $= -\sum_{k=1}^K \frac{|C_k|}{|D|} log\frac{|C_k|}{|D|} + \sum_{i=1}^n \frac{|D_i|}{|D|} \sum_{k=1}^K \frac{|D_{ik}|}{|D_i|} log\frac{|D_{ik}|}{|D_i|}$

1.2 information gain ratio

> **Def** Normalization: tackling the possibility of a overwhelming variable set

> **Equ** $g_R(D,A) = \frac{g(D,A)}{H_A(D)}$
> $= \frac{g(D,A)}{-\sum_{i=1}^{n}\frac{|D_i|}{|D|}log\frac{|D_i|}{|D|}}$

1.3 Gini coefficient

> **Def** purity of the data set

> **Equ** $Gini(D) = \sum_{i \neq j} PiPj$
> $\because binaryTree$
> $= \sum_{k=1}^{K} P_k(1-P_k))$
> $= 1 - \sum_{k=1}^{K}(P_k)^2$
> $= 1 - \sum_{k=1}^{K}(\frac{|C_k|}{|D|})^2$

> **Equ** $Gini(D,A) = \frac{|D_1|}{|D|}Gini(D1) + \frac{|D_2|}{|D|}Gini(D2)$

2.Generation

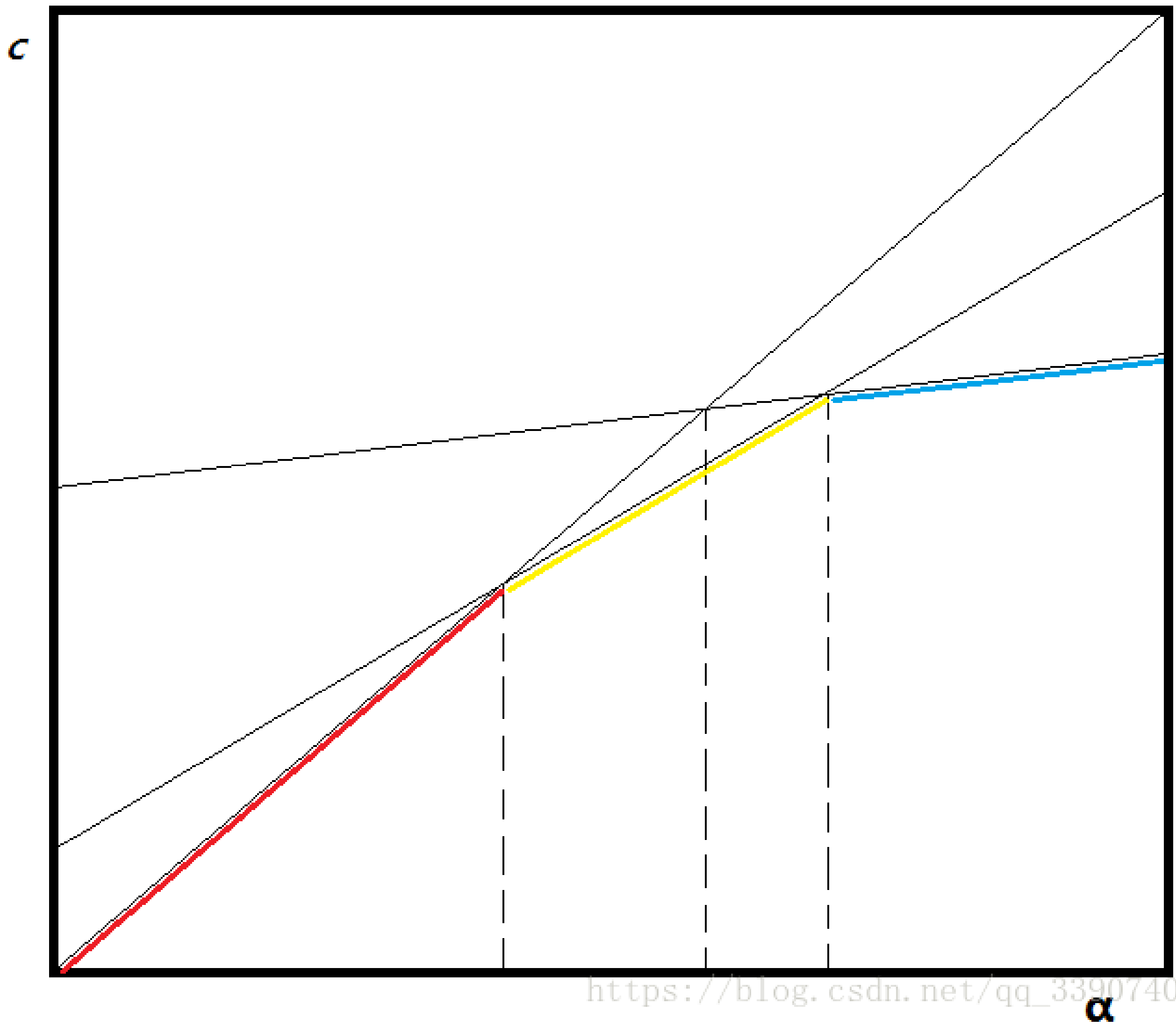
> **Def**
> $ID3$ ---- information gain
> $C4.5$ ---- information gain ratio
> $1)$ calculate each gain , choose the maximum gain.
> $2)$ divide recursively until (subtree in same class) or (gain $\leq$ threshold)

> $CART$ ---- Gini coefficient
> $1)$ choose the min $Gini(D,A_i)$ as the optimal segmentation point
> $2)$ divide recursively until (each feature traversed) or (subtree in same class)

3.Pruning

> **Def** alleviate degree of overfitting
> **loss functon** a function that maps an event or values of one or more variables onto a real number intuitively representing some “cost” associated with the event.

> **Equ** leaf node number $|T|$, class number N_t, parameter α
> $C_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha\,|T|$
> $= -\sum_{t=1}^{|T|} \sum_{k=1}^{K} N_{tk} log \frac{N_{tk}}{N_t} + \alpha\,|T|$
> decreasing loss ---- need pruning



> $CART\ Pruning$
> Situation: pruning without a given parameter α
> In the way of loss function, leafy trees benefit from little α and the other way around.
> Pruning happens when loss function of a inner node acting as a root equals to the one acting as a leaf.
> $C_\alpha(t) = C(t) + \alpha$
> $C_\alpha(T_t) = C(T_t) + \alpha\,|T|$
> $\because C_\alpha(t) = C_\alpha(T_t)$
> $\therefore \alpha = \frac{C(t)-C(T_t)}{|T|-1}$
> $1)$ each inner node has an α
> $2)$ acquire an ascending set of α , and a set of subtrees accordingly
> $3)$ Cross Validation: test the subtree set, select the one with the highest accuracy rate