Classification Tree

Let us now look at various issues associated with construction of classification trees.

Let T be a tree with $\tilde{T} = \{t_1, \ldots, t_M\}$ as the set of terminal nodes.

Let $\pi_{j(t)} \in \{\pi_1, \ldots, \pi_c\}$ denote one of the class labels, that is associated with terminal node t

A classification tree consists of T, $\tilde{T}$, $\{\pi_{j(t)}, t \in \tilde{T}\}$ and a partition $\{U(t) : t \in \tilde{T}\}$.

Let the learning sample be $\mathcal{L} = \{(\underset{\sim}{x}_i, y_i) : i = 1(1)N\}$
(y_is are the class labels)

$\forall t \in T$, define

$N(t) = \#$ of sample patterns in $\mathcal{L} \ni \underset{\sim}{x}_i \in U(t)$

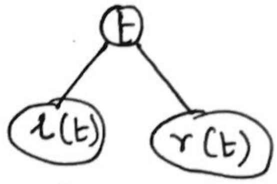
$N_j(t) = \#$ - - - - - - - $\ni \underset{\sim}{x}_i \in U(t)$ and $y_i = \pi_j$

Clearly, $\sum_j N_j(t) = N(t)$ & $\sum_{t \in \tilde{T}} N(t) = N$

$$p(t) = \frac{N(t)}{N} \text{ ; estimate of } P(\underset{\sim}{X} \in U(t)) \text{ based on } \mathcal{L}$$

$$p(\pi_j | t) = \frac{N_j(t)}{N(t)} \text{ ; estimate of } P(Y = \pi_j \mid \underset{\sim}{X} \in U(t))$$

Recall that $\forall t \in T$



let $t_L = l(t)$ & $t_R = r(t)$

$$p_L^t = \frac{p(t_L)}{p(t)} \text{ ; estimate of } P(\underset{\sim}{X} \in U(t_L) \mid \underset{\sim}{X} \in U(t))$$

$$p_R^t = \frac{p(t_R)}{p(t)} \text{ ; estimate of } P(\underset{\sim}{X} \in U(t_R) \mid \underset{\sim}{X} \in U(t))$$

<u>Rule for label assignment</u> (class label assignment)

Assign label π_j to node t if

$$p(\pi_j \mid t) = \max_i p(\pi_i \mid t)$$

(i.e. majority voting inside the partition)

<u>Splitting rules</u>

Splitting rules at internal nodes are based on "node impurity" measures.

In general "node impurity" for any node $t \in T$ is defined as

$$Imp(t) = \phi\left(p(\pi_1 \mid t), p(\pi_2 \mid t), \dots, p(\pi_c \mid t)\right)$$

<u>Note:</u>

Such a "node impurity" would be maximum if $p(\pi_j \mid t) = \frac{1}{c} \;\; \forall j$ (i.e. equal representation of all classes at t)

and minimum if $\left.\begin{array}{l} p(\pi_j \mid t) = 1 \text{ for some } j \\ \& \; p(\pi_i \mid t) = 0 \;\; \forall i \neq j \end{array}\right\}$ it's a pure node

Examples of "node impurity" measures

(i) Gini Index $(t) = \sum_{\substack{i,j \\ i \neq j}} p(\pi_i|t)\, p(\pi_j|t)$

(If $c=2$ (two-class problem)

Gini Index $(t) = \sum_{i=1}^{2} \sum_{\substack{j=1 \\ j\neq i}}^{2} p(\pi_i|t)\, p(\pi_j|t)$

$= p(\pi_1|t)\, p(\pi_2|t) + p(\pi_2|t)\, p(\pi_1|t)$

$= p(1-p) + (1-p)p \quad \left(p(\pi_1|t) = p \right)$

$= 2p(1-p)$

max if $p = \frac{1}{2}$ (i.e. $\frac{1}{c}$)

min if p or $1-p = 0/1$)

(ii) Misclassificate error rate at node t

$$= \frac{1}{N(t)} \sum_{i:\, \underset{\sim}{x}_i \in U(t)} I\left(y_i \neq \pi_{j(t)} \right)$$

$j(t) = \arg\max_i p(\pi_i|t)$

$$= \frac{1}{N(t)} \left(N(t) - N_{j(t)}(t) \right)$$

$$= 1 - \frac{N_{j(t)}(t)}{N(t)} = 1 - p(\pi_j|t)$$

(iii) Cross-entropy or deviance

$$-\sum_i p(\pi_i|t) \operatorname{Log} p(\pi_i|t)$$

<u>Remark</u>: The above measures are for node impurity. We can define tree impurity as

$$\mathrm{Imp}(T) = \sum_{t \in \tilde{T}} p(t)\,\mathrm{Imp}(t)$$

How to split a node?

This is by far the most important question!!

Consider a split using variable x_K at level l

say, $s_K^l = \{\underset{\sim}{x} : x_K < l\}$

We define a measure of "goodness of split" at node t as the change in impurity f^n. For the split s_K^l at t, this is

$$\Delta\,\mathrm{Imp}(s_K^l, t) = \mathrm{Imp}(t) - \left(p_L^t\,\mathrm{Imp}(t_L) + p_R^t\,\mathrm{Imp}(t_R)\right)$$

(recall that $p_L^t = P(\underset{\sim}{X} \in U(t_L) \mid \underset{\sim}{X} \in U(t))$
& $p_R^t = P(\underset{\sim}{X} \in U(t_R) \mid \underset{\sim}{X} \in U(t))$).

We need to find $(K^*, l^*) \ni \Delta\,\mathrm{Imp}(s_K^l, t)$ is maximized over all $K = 1(1)p$ (dimension of feature vector) and l is allowed take one of a finite # of values within the range of possible values of the choosen feature variable.

The above "goodness of split" based approach is used to split nodes starting with the root node.

When to stop splitting?

We do not split a node if the change in the impurity due to split is less than a prescribed threshold

OR

Grow the tree till the terminal nodes are all pure (all ~~[illegible]~~ patterns belonging to the partition have same class label) and then apply prunning of the tree.

What is prunning? How to prune a classification tree?

There are various approaches of prunning. We discuss here 2 important approaches.

Prunning of a grown tree after applying splitting of nodes basically means cutting branches of the tree to get a subtree of the original tree.

Cost - Complexity prunning

Let $r(t) = 1 - \max_i p(\pi_i | t)$.

$\uparrow$ estimate of prob of misclassification at node t

define
$$R(t) = p(t)\, r(t)$$

Estimate of overall misclassification rate of the tree classifier is
$$R(T) = \sum_{t \in \tilde{T}} p(t)\, r(t) = \sum_{t \in \tilde{T}} R(t).$$

If α denotes the cost of complexity per terminal node then define

$R_\alpha(t) = R(t) + \alpha$ as cost-complexity criterion for node t

& $R_\alpha(T) = \sum_{t \in \tilde{T}} (R(t) + \alpha)$ (α : tuning parameter a trade-off parameter)

$$= \sum_{t \in \tilde{T}} R(t) + \alpha |\tilde{T}|$$

where $|\tilde{T}|$: cardinality of terminal node set $\tilde{T}$.

Cost-complexity prunning approach : Starting from T_0 (a pure tree, i.e. a tree having all terminal nodes as pure), find the subtree T_α (for a fixed α) $\ni$ $R_\alpha(T)$ is minimum.

Weakest link pruning approach

Let T_t be subtree with root t

& $\{t\}$ is subbranch of T_t consisting of a single node t

Let $R_\alpha(t) = R(t) + \alpha = r(t)\, p(t) + \alpha$

(as defined earlier)

& for T_t; $R_\alpha(T_t) = R(T_t) + \alpha|\tilde{T}_t|$

Now $R_\alpha(T_t) < R_\alpha\{t\}$ (i.e. T_t has smaller cost complexity than $\{t\}$).

if $R_{\cancel{\alpha}}(T_t) + \alpha|\tilde{T}_t| < R(t) + \alpha$

i.e. if $\alpha(|\tilde{T}_t| - 1) < R(t) - R(T_t)$

i.e. if $\alpha < \dfrac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$ — (*)

Note that as $\alpha \uparrow$ equality is achieved at (*) and T_t and $\{t\}$ have same cost-complexity and $\{t\}$ is preferred to T_t; i.e. T_t can be prunned at t

We define

$$g(t) = \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$$

as the "strength of link" at node t

Find $t^* \ni g(t^*)$ is min and then prune the tree at this "weakest-link", i.e. at t^*.

At the next step start with the prunned subtree and find the weakest link and prune again.

Continue to reach the root through prunning to get the sequence of prunned subtrees

$$T_1 \supset T_2 \supset \cdots \supset \{t_1\}$$

org root

Under this approach, the prunned subtree with min $R_\alpha(T)$ is the best prunned tree.

Growing a regression tree

How to split nodes?

$\mathcal{L} = \{ (\underset{\sim}{x}_i, y_i) : i = 1(1)N \}$

Consider a split variable j and a split pt t

$R_1(j,t) = \{ \underset{\sim}{x} : x_j < t \}$ & $R_2(j,t) = \{ \underset{\sim}{x} : x_j \geq t \}$

Set the criterion f^n as

$$\underset{j,\,t}{\text{Min}} \left[\min_{c_1} \sum_{i : \underset{\sim}{x}_i \in R_1(j,t)} (y_i - c_1)^2 + \min_{c_2} \sum_{i : \underset{\sim}{x}_i \in R_2(j,t)} (y_i - c_2)^2 \right] \quad -(*)$$

Note that for a fixed j & t

$$\min \rightarrow \hat{c}_1 = \text{average}\left(y_i \mid \underset{\sim}{x}_i \in R_1(j,t) \right)$$

$$\& \ \hat{c}_2 = \text{average}\left(y_i \mid \underset{\sim}{x}_i \in R_2(j,t) \right)$$

Find (j^*, t^*) for the optimum split at a node $\ni$

$$\sum_{i : \underset{\sim}{x}_i \in R_1(j,t)} (y_i - \hat{c}_1)^2 + \sum_{i : \underset{\sim}{x}_i \in R_2(j,t)} (y_i - \hat{c}_2)^2 \quad \text{is minimised}$$

here j varies over all $j = 1(1)p$ (feature vector dim)

& t in the grid of the j^{th} variable

When to stop splitting?

(I) Split tree node only if the decrease in sum of squares residuals due to split exceeds some pre-assigned threshold

OR

(II) Std dev of responses at terminal nodes is low (below a threshold)

OR

(III) Grow a large tree and stop splitting if the node size (# of patterns reaching that node) reaches a low threshold level and apply prunning.

Regression tree prunning

$\forall t \in \tilde{T}$ or T define $\hat{c}_t = \frac{1}{N(t)} \sum_{\underset{\sim}{x}_i \in U(t)} y_i$

$N(t)$: # of $\underset{\sim}{x}_i \in U(t)$

Impurity measure:

Mean square error $\forall$ obsn in $U(t)$ from $\hat{c}_t$

$$Q_t = \frac{1}{N(t)} \sum_{i : \underset{\sim}{x}_i \in U(t)} (y_i - \hat{c}_t)^2$$

Define cost complexity f^n as

$$C_\alpha(T) = \sum_{t \in \tilde{T}} N(t)\, Q_t + \alpha |\tilde{T}|$$

α: tuning parameter which governs the trade-off bet^n tree size and goodness of fit

$|\tilde{T}|$: cardinality of $\tilde{T}$.

Pruning can be done based on $C_\alpha(T)$. For a fixed α, let T_α be a subtree of T obtained by pruning T. We try to find

$$T_\alpha \subset T \ni T_\alpha \text{ minimizes } C_\alpha(T)$$

for a fixed α.

(Note: larger the α smaller in the opt T_α)

<u>Weakest link pruning</u>: Collapse internal nodes (to prune) that produces the smallest per node increase in

$$\sum_{t \in \tilde{T}} N(t)\, Q(t)$$ and continue till root node is reached.

Select the optimal from the pruned subtree sequence.