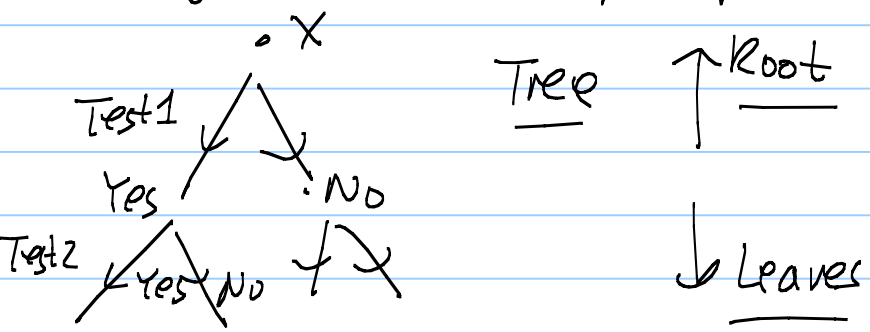
Spring 2014

(1)

Note Title 11/7/2006

Decision Trees (New Topic)

Game of Twenty Questions
Apply a series of tests to the input pattern



Notation:

Set of Classified Data
$\{(x_\alpha, \omega_\alpha) : \alpha \in \Lambda\}$

Set of Tests $\{T_j : j \in \Phi\}$

Each test has response "T" or "F"
$T_j(x_\alpha) \in \{T, F\}$

Tree nodes $\{\mu\}$
Root node μ_0 at top of tree.
Each node either has two child nodes, or is a leaf node.

Each node $\{\mu\}$ has a test T_μ. Its child node μ_1 is for data x_α s.t. $T_\mu(x_\alpha) = T$
μ_2 " " " " " $T_\mu(x_\alpha) = F$.

(2)

Decision Tree Notation

Define the data that gets to node μ recursively.

$$\Lambda_{\mu 1} = \{ x_a \in \Lambda_\mu \text{ s.t. } T_\mu(x_a) = T \}$$
$$\Lambda_{\mu 2} = \{ x_a \in \Lambda_\mu \text{ s.t. } T_\mu(x_a) = F \}$$

The root node contains all data $\Lambda_{\mu_0} = \Lambda$.

- Distribution of data at node μ.

$$P_\mu(\omega_j) = \frac{1}{|\Lambda_\mu|} \sum_{a \in \Lambda_\mu} \delta_{\omega_a \omega_j}$$

$$\sum_{j=1}^{M} P_\mu(\omega_j) = 1 \qquad M \text{ is no. classes}$$

- Define an impurity measure (entropy) for node μ

$$I(\mu) = -\sum_j P_\mu(\omega_j) \log P_\mu(\omega_j)$$

Note: if a node is pure, then all data in it belongs to one class.

Intuition: Design a tree so that the leaf nodes are pure — yield good classification

(3)

Iterative Design.

- Initialize tree with the root node only. (so it is a leaf node).
- For all leaf nodes, calculate the maximal decrease in impurity by searching over all the tests.
- Expand the leaf node with maximal decrease and add its child nodes to the tree.

Decrease in impurity at node μ due to test T_j.

$$\Lambda^j_{\mu,1} = \{x \in \Lambda_\mu : \text{s.t. } T_\mu(x) = T\}$$
$$\Lambda^j_{\mu,2} = \{x \in \Lambda_\mu : \text{s.t. } T_\mu(x) = F\}.$$

Decrease in entropy $\Delta^j I(\mu) = I(\mu) - I(\mu^j_1, \mu^j_2)$.

where $I_{\mu^j_1,\mu^j_2} = \frac{|\Lambda_{\mu^j_1}|}{|\Lambda_\mu|} I(\mu^j_1) + \frac{|\Lambda^j_{\mu,2}|}{|\Lambda_\mu|} I(\mu^j_2)$,

Hence, for all leaf node μ calculate $\max_j I^j_{\mu_1,\mu_2}$. Select the leaf node μ and test T_j which achieves this maximum.

Greedy Strategy

- Start at root node.
- Expand root node with test that maximizes the decrease in impurity — or maximizes the gain in purity.
- Repeat with leaf nodes. until each leaf node is pure.

Time Complexity: Learning algorithm is

$O(|\phi||N|\{\log|N|\}^2)$ $|\phi|$ = no. of tests.

Run Time $O(\log|N|)$. Very Rapid

Notes: the design strategy is very greedy. There may be a shorter tree if you learn the tree by searching over a sequence of tests.

The number of children (2) is arbitrary. You can extend the approach to having three, or more, children

(5)

There are alternative impurity measures (e.g. the Gini index)

$$I(\mu) = \sum_{i,j\,(i\neq j)} P_\mu(\omega_i) P_\mu(\omega_j) = 1 - \sum_j P_\mu^2(\omega_j)$$

Expanding the tree until all nodes are pure risks overgeneralizing. It will give perfect performance on the training dataset, but will usually cause errors on the test dataset.

Better to stop splitting the data when the impurity reaches a positive threshold, i.e. set a node to be a leaf if $I(\mu) \leq \beta$ — threshold

Then at each leaf, classify data by majority vote.

Cross Validation Strategy: learn the decision tree with different impurity thresholds β. Select the tree, and hence the β, which has best validation (consistency between training & test datasets.