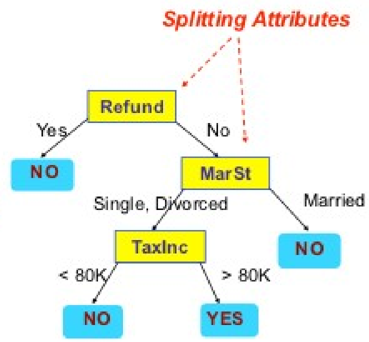
Classification trees

	categorical	categorical	continuous	class
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

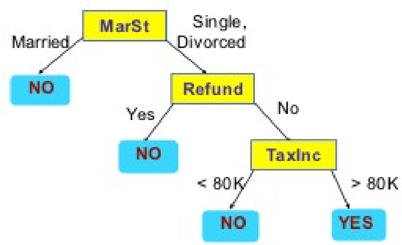


Model: Decision Tree

Classification trees

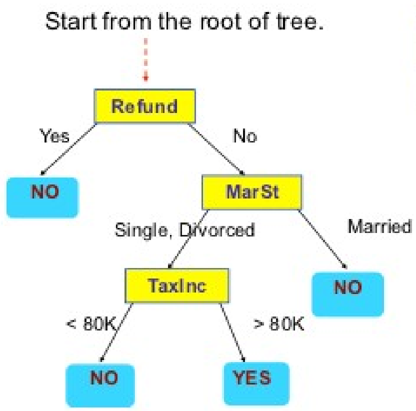
categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



There could be more than one tree that fits the same data!

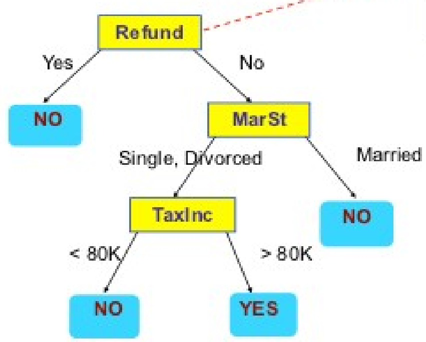
Applying a decision tree rule



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

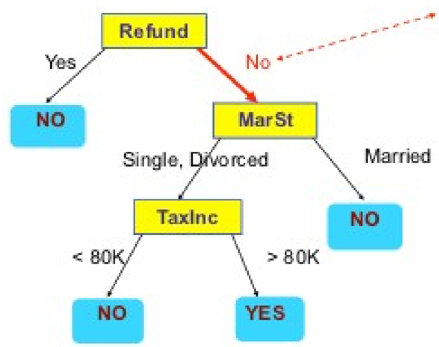
Applying a decision tree rule



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

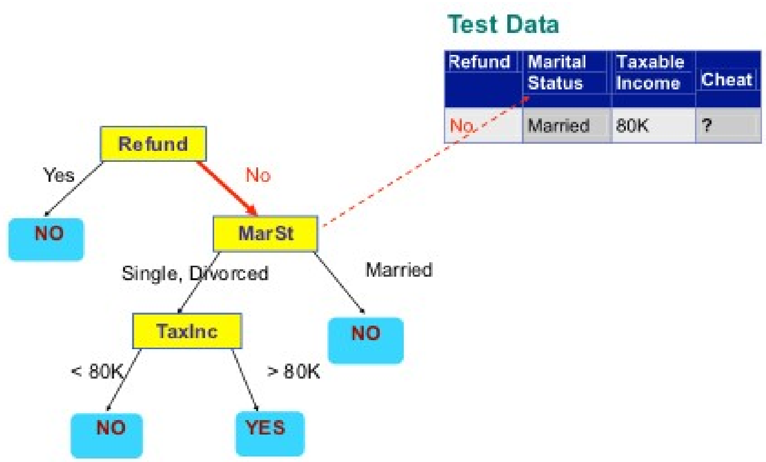
Applying a decision tree rule



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

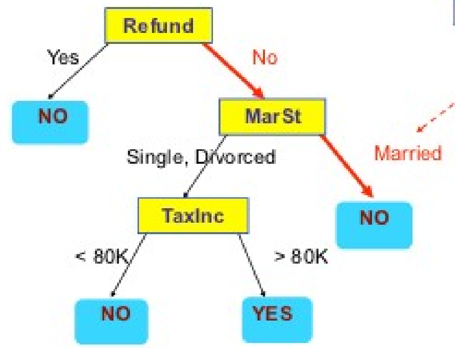
Applying a decision tree rule



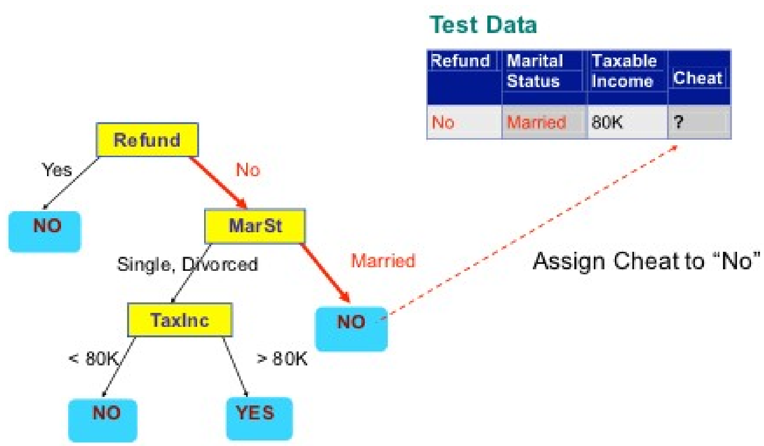
Applying a decision tree rule

Test Data

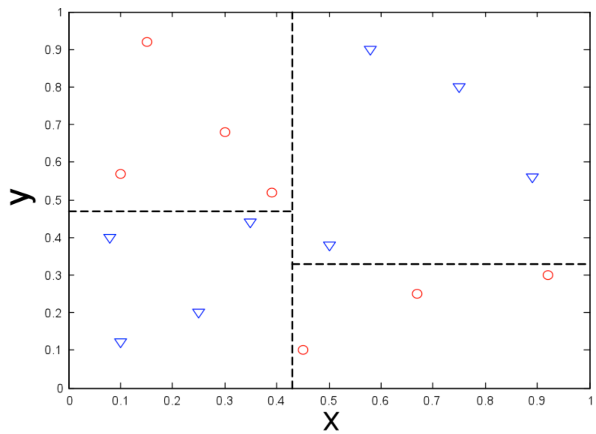
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Applying a decision tree rule



Decision boundary for tree



Decision boundary for tree

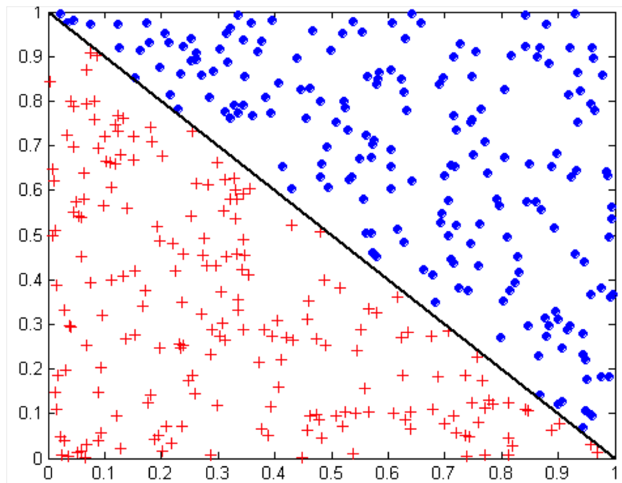


Figure : Trees have trouble capturing structure not parallel to axes

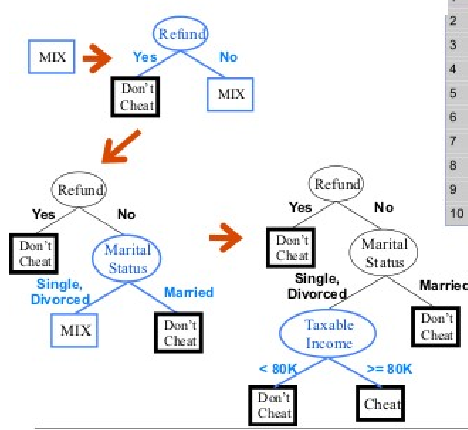
Learning the tree

Hunt's algorithm (generic structure)

- Let D_t be the set of training records that reach a node t
- If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t.
- If $D_t = \emptyset$, then t is a leaf node labeled by the default class, y_d.
- If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.
- This splitting procedure is what can vary for different tree learning algorithms ...

Learning the tree

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Learning the tree

Issues

Greedy strategy: Split the records based on an attribute test that optimizes certain criterion.

What is the best split: What criterion do we use? Previous example chose first to split on `Refund` ...

How to split the records: Binary or multi-way? Previous example split `Taxable Income` at $\geq 80K$ ...

When do we stop? Should we continue until each node if possible? Previous example stopped with all nodes being completely homogeneous ...

Different splits: ordinal / nominal

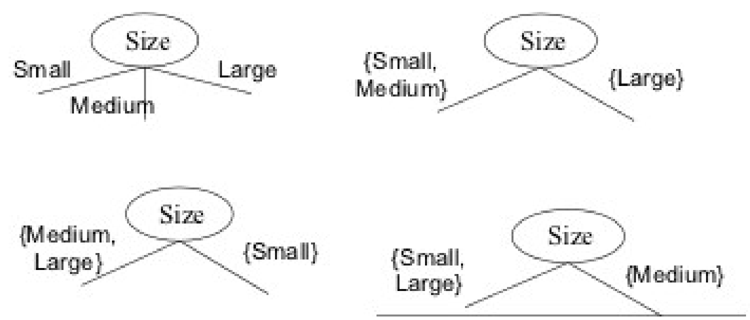


Figure : Binary or multi-way?

Different splits: continuous

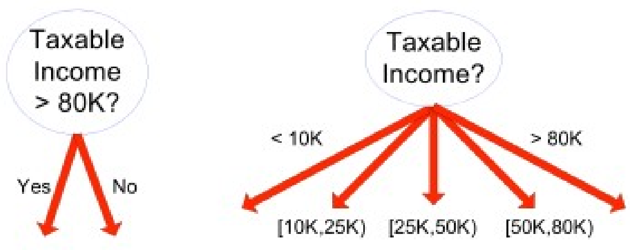


Figure : Binary or multi-way?

Choosing a variable to split on

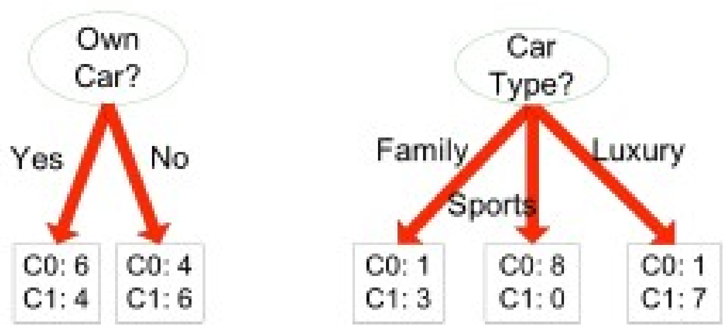


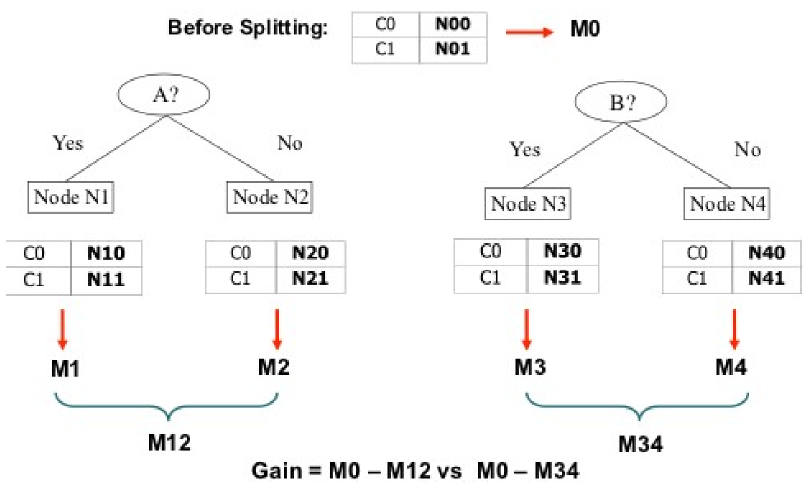
Figure : Which should we start the splitting on?

Learning the tree

Choosing the best split

- Need some numerical criterion to choose among possible splits.
- Criterion should favor *homogeneous or pure* nodes.
- Common cost functions:
  - Gini Index
  - Entropy / Deviance / Information
  - Misclassification Error

Choosing a variable to split on



Learning the tree

GINI Index

- Suppose we have k classes and node t has frequencies $p_t = (p_{1,t}, \dots, p_{k,t})$.
- Criterion

$$GINI(t) = \sum_{(j,j') \in \{1,\dots,k\}: j \neq j'} p_{j,t} p_{j',t} = 1 - \sum_{j=1}^{l} p_{j,t}^2.$$

- Maximized when $p_{j,t} = 1/k$ with value $1 - 1/k$
- Minimized when all records belong to a single class.
- Minimizing *GINI* will favour *pure* nodes ...

Learning the tree

Gain in GINI Index for a potential split

- Suppose t is to be split into j new child nodes $(t_l)_{1 \leq l \leq j}$.
- Each child node has a count n_l and a vector of frequencies $(p_{1,t_l}, \dots, p_{k,t_l})$. Hence they have their own GINI index, $GINI(t_l)$.
- The gain in GINI Index for this split is

$$\text{Gain}(GINI, t \to (t_l)_{1 \leq l \leq j}) = GINI(t) - \frac{\sum_{l=1}^{j} n_l GINI(t_l)}{\sum_{l=1}^{j} n_l}.$$

- Greedy algorithm chooses the biggest gain in GINI index among a list of possible splits.

Learning the tree

Entropy / Deviance / Information

- Suppose we have k classes and node t has frequencies $p_t = (p_{1,t}, \dots, p_{k,t})$.
- Criterion

$$H(t) = -\sum_{j=1}^{k} p_{j,t} \log p_{j,t}$$

- Maximized when $p_{i,t} = 1/k$ with value $\log k$
- Minimized when one class has no records in it.
- Minimizing entropy will favour *pure* nodes ...

Learning the tree

Gain in entropy for a potential split

- Suppose t is to be split into j new child nodes $(t_l)_{1 \leq l \leq j}$.
- Each child node has a count n_l and a vector of frequencies $(p_{1,t_l}, \dots, p_{k,t_l})$. Hence they have their own entropy $H(t_l)$.
- The gain in entropy for this split is

$$\text{Gain}(H, t \to (t_l)_{1 \leq l \leq j}) = H(t) - \frac{\sum_{l=1}^{j} n_l H(t_l)}{\sum_{l=1}^{j} n_l}.$$

- Greedy algorithm chooses the biggest gain in H among a list of possible splits.

Learning the tree

Stopping training

- As trees get deeper, or if splits are multi-way the number of data points per leaf node drops very quickly.
- Trees that are too deep tend to overfit the data.
- A common strategy is to "prune" the tree by removing some internal nodes.

Learning the tree

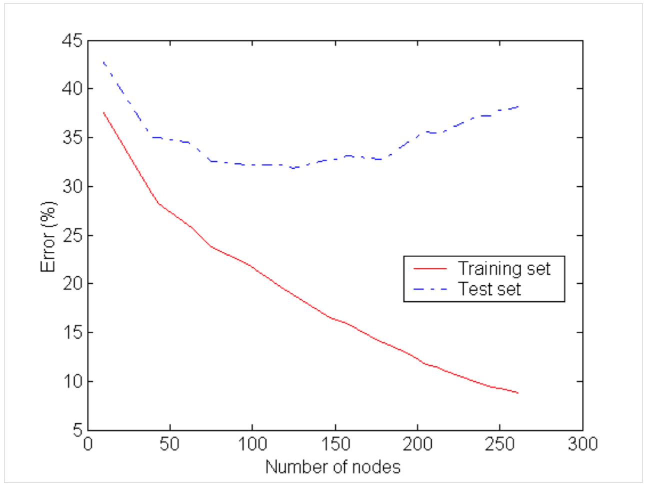


Figure : Underfitting corresponds to the left-hand side, overfit to the right

Learning the tree

Statistics 202: Data Mining

©Jonathan Taylor

Cost-complexity pruning (`tree` library)

- Given a criterion Q like H or $GINI$, we define the cost-complexity of a tree with terminal nodes $(t_j)_{1 \leq j \leq m}$

$$C_\alpha(T) = \sum_{j=1}^{m} n_j Q(t_j) + \alpha m$$

- Given a large tree T_L we might compute $C_\alpha(T)$ for any subtree T of T_L.
- The optimal tree is defined as

$$\hat{T}_\alpha = \operatorname*{argmin}_{T \leq T_L} C_\alpha(T).$$

- Can be found by "weakest-link" pruning. See *Elements of Statistical Learning* for more ...