Decision Tree

Li, Jia

DSAA 5002
HKUST Guangzhou

2025 Fall

Sep 1

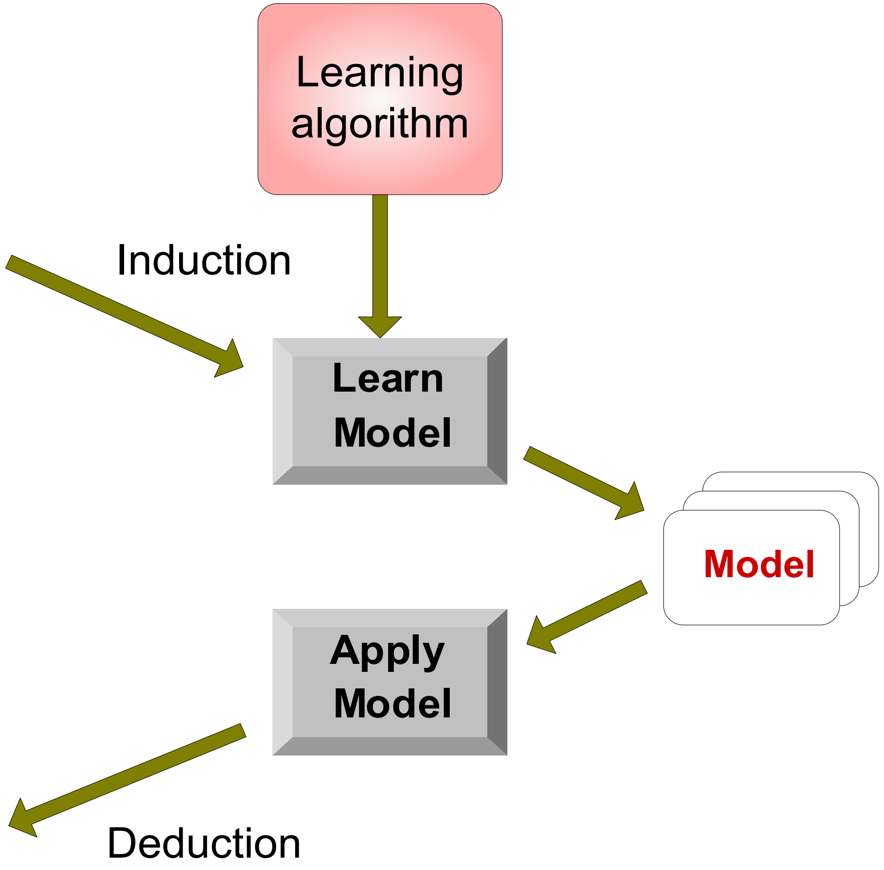
Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

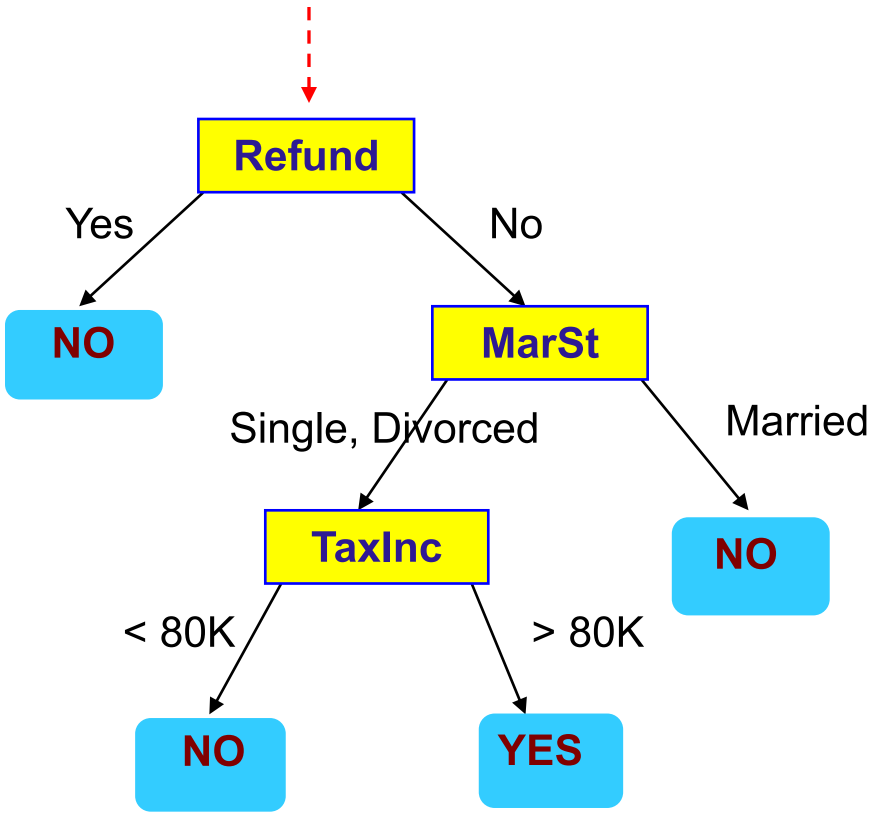


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Decision Tree

Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Tree Induction

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

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

- Depends on number of ways to split
  - 2-way split
  - Multi-way split

How to determine the Best Split

- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

C0: 5	C0: 9
C1: 5	C1: 1
Non-homogeneous,	**Homogeneous,**
High degree of impurity	**Low degree of impurity**

Measures of Node Impurity

- Gini Index

- Entropy

Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Maximum ($1 - 1/n_c$) when records are equally distributed among all classes, implying least interesting information
- Minimum (0.0) when all records belong to one class, implying most interesting information

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Examples for computing GINI

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Gini = 1 – $P(C1)^2$ – $P(C2)^2$ = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

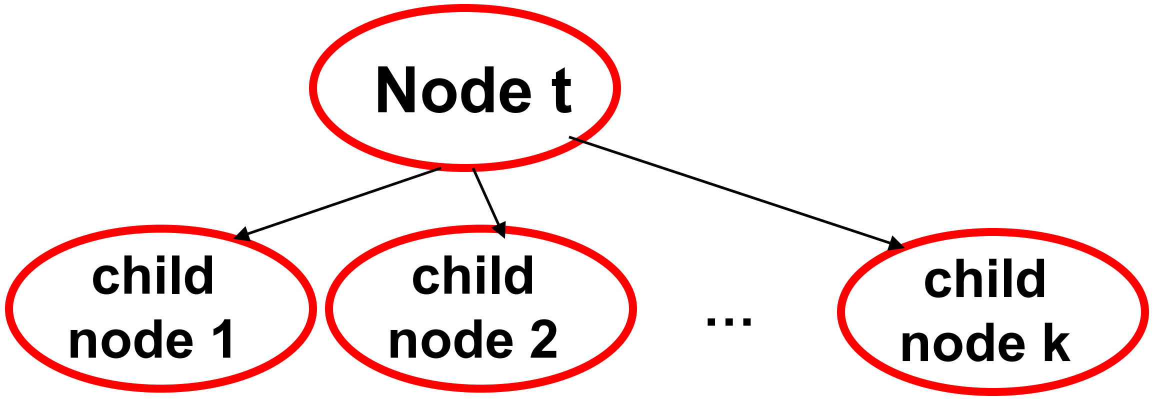
Splitting Based on GINI

- Used in CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i,

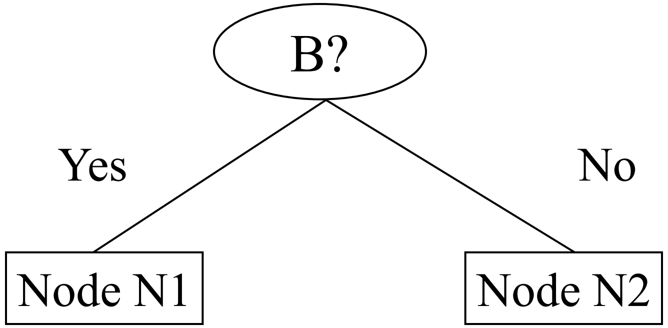
n = number of records at node p.



Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.

	Parent
C1	**6**
C2	**6**
Gini = 0.500	



Gini(N1)
$= 1 – (5/7)^2 – (2/7)^2$
$= 0.408$

Gini(N2)
$= 1 – (1/5)^2 – (4/5)^2$
$= 0.320$

	N1	N2
C1	**5**	**1**
C2	**2**	**4**
Gini=0.371		

Gini(Children)
$= 7/12 * 0.408 + 5/12 * 0.320$
$= 0.371$

Alternative Splitting Criteria based on INFO

- Entropy at a given node t:

$$\boxed{Entropy(t) = -\sum_{j} p(j \mid t) \log p(j \mid t)}$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Measures homogeneity of a node.
  - Maximum ($\log n_c$) when records are equally distributed among all classes implying least information
  - Minimum (0.0) when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

Examples for computing Entropy

$$Entropy(t) = -\sum_j p(j\,|\,t)\log_2 p(j\,|\,t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Splitting Based on INFO...

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions;

n_i is number of records in partition i

  - Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
  - Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Tree Induction

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

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

Occam's Razor

- Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
- For complex models, there is a greater chance that it was fitted accidentally by errors in data
- Therefore, one should include model complexity when evaluating a model

Metrics for Performance Evaluation

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Confusion Matrix

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Metrics for Performance Evaluation

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

Errors

Next, we will provide a theoretical explanation about overfitting.

Given a classifier h, define its error on S — denote as $err_s(h)$ to be:

$$err_S(h) = \frac{|\{(\boldsymbol{x}, y) \in S \mid h(\boldsymbol{x}) \neq y\}|}{|S|}.$$

namely, the percentage of objects in S whose labels are incorrectly predicted by h.

Remark:

- $err_s(h)$ is often called the empirical error of h.
- $err_d(h)$ is often called the generalization error of h.

Generalization Theorem

Let H be the set of classifiers that can possibly be returned. The following statement holds with probability at least 1 – δ (where 0< δ ≤1): for any h ∈ H

$$err_{\mathcal{D}}(h) \leq err_S(h) + \sqrt{\frac{\ln(1/\delta) + \ln|H|}{2|S|}}.$$

We should:

- Look for a decision tree that is both accurate on the training set and small in size;
- Increase the size of S as much as possible.

Hoeffding Bounds (Optional)

Let $X_1, \ldots, X_n$ be independent Bernoulli random variables satisfying $R_r[X_i = 1] = p$ for all i. Set $s = \sum_{i=1}^{n} X_i$. Then, for any $0 \leq \alpha \leq 1$:

$$\begin{aligned} \textbf{\textit{Pr}}[s/n > p + \alpha] &\leq e^{-2n\alpha^2} \\ \textbf{\textit{Pr}}[s/n < p - \alpha] &\leq e^{-2n\alpha^2}. \end{aligned}$$

Union Bound (Optional)

Let $E_1, \ldots, E_n$ be n arbitrary events such that event E_i happens with probability P_i . Then,

$$\boldsymbol{Pr}[\text{at least one of } E_1, \ldots, E_n \text{ happens}] \leq \sum_{i=1}^{n} p_i.$$

Proof of the Generalization Theorem (Optional)

For a classifier h ∈ H. Let S be the training set with n=|S|. For each i ∈ [1,n], define X_i=1 if the i-th object in S in incorrectly predicted by h, or 0 otherwise. We have

$$err_S(h) = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Since each object in S is drawn from D independently, for every i:

$$\boldsymbol{Pr}[X_i = 1] = err_{\mathcal{D}}(h).$$

Proof of the Generalization Theorem (Optional)

By Hoeffding bounds, we get:

$$\boldsymbol{Pr}[err_S(h) < err_{\mathcal{D}}(h) - \alpha] \quad \leq \quad e^{-2n\alpha^2}$$

Which is at most $\delta/|H|$ by setting $e^{-2n\alpha^2} = \delta/|H|$, namely

$$\alpha = \sqrt{\frac{\ln(1/\delta) + \ln|H|}{2n}}.$$

We say h fails if $err_s(h) < err_D(h)$ - α .

Proof of the Generalization Theorem (Optional)

The above analysis shows that each classifier in H fails with probability at most δ /|H|. By the Union Bound, the probability that at least one classifier in H fails is at most δ. Hence, the probability that no classifiers fail is at least 1 – δ.

Slides Credit

Many slides are adopted from Lecture Notes for Chapter 4 Introduction to Data Mining By Tan, Steinbach, Kumar

Other references:

[1] Yufei Tao. Note 1 in Data Mining and Knowledge Discovery.