Classification and Prediction

——Classification by Decision Tree——

徐华

清华大学 计算机系 智能技术与系统国家重点实验室

xuhua@tsinghua.edu.cn

Classification and Prediction

- Basic Concepts
- Issues Regarding Classification and Prediction
- Decision Tree
- Bayesian Classification
- Neural Networks
- Support Vector Machine
- K-Nearest Neighbor
- Associative classification
- Classification Accuracy

Training Dataset

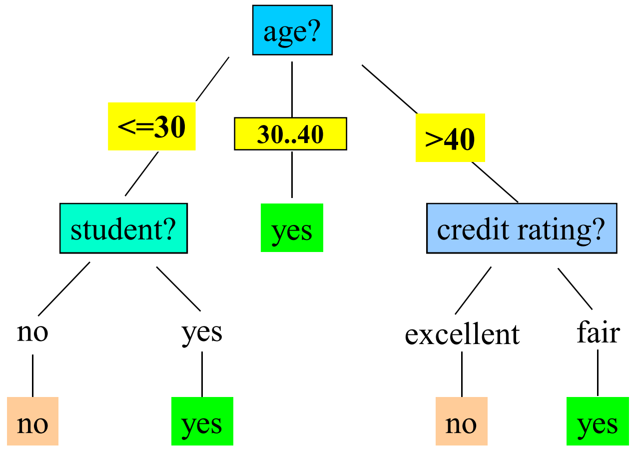
- This follows an example from Quinlan's ID3

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Training Dataset

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
>40	medium	yes	fair	yes
>40	medium	no	excellent	no
31…40	high	no	fair	yes
31…40	low	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes

Output: A Decision Tree for "buys_computer"



Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain(信息增益)
- S contains s_i tuples of class C_i for i = {1, …, m}
- **information** measures info required to classify any arbitrary tuple

$$I(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

- **Entropy（熵）** of attribute A with values $\{a_1,a_2,\ldots,a_v\}$

$$E(A)=\sum_{j=1}^{v} \frac{s_{1j}+\ldots+s_{mj}}{s} I(s_{1j},\ldots,s_{mj})$$

- **information gained（信息增益）** by branching on attribute A

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A)$$

7

Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- I(p, n) = I(9, 5) =0.940
- Compute the entropy for age:

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$ **means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence**

$$Gain(age) = I(p,n) - E(age) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

8

Computing Information-Gain for Continuous-Value Attributes

- **Let attribute A be a continuous-valued attribute**
- **Must determine the *best split point* for A**
  - **Sort the value A in increasing order**
  - **Typically, the midpoint between each pair of adjacent values is considered as a possible *split point***
    - **$(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}**
  - **The point with the *minimum expected information requirement* for A is selected as the split-point for A**
- **Split:**
  - **D1 is the set of tuples in D satisfying A ≤ split-point, and D2 is the set of tuples in D satisfying A > split-point**

Gain Ratio for Attribute Selection (C4.5)

- **Information gain measure is biased towards attributes with a large number of values**
- **C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)**

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

  - **GainRatio(A) = Gain(A)/SplitInfo(A)**
- **Ex.** $SplitInfo_A(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 0.926$
  - **gain_ratio(income) = 0.029/0.926 = 0.031**
- **The attribute with the maximum gain ratio is selected as the splitting attribute**

Gini index (CART, IBM IntelligentMiner)

- **If a data set *D* contains examples from *n* classes, gini index, *gini*(*D*) is defined as**

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class *j* in *D*

- **If a data set *D* is split on A into two subsets D_1 and D_2, the *gini* index *gini*(*D*) is defined as**

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- **Reduction in Impurity:**

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- **The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)**

Gini index (CART, IBM Intelligent Miner)

- **Ex. D has 9 tuples in buys_computer = "yes" and 5 in "no"**

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- **Suppose the attribute income partitions D into 10 in D_1: {low, medium} and 4 in D_2**

$$gini_{income \in \{low, medium\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2)$$

but $gini_{\{medium,high\}}$ is 0.30 and thus the best since it is the lowest

- **All attributes are assumed continuous-valued**
- **May need other tools, e.g., clustering, to get the possible split values**
- **Can be modified for categorical attributes**

Comparing Attribute Selection Measures

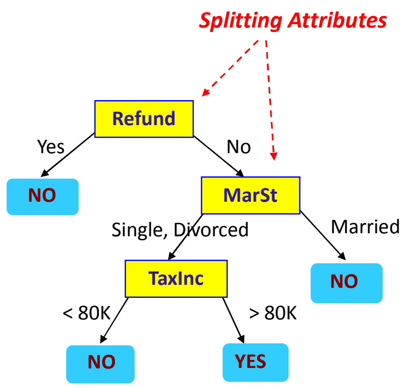
- The three measures, in general, return good results but
  - Information gain:
    - biased towards multivalued attributes
  - Gain ratio:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - Gini index:
    - biased to multivalued attributes
    - has difficulty when the number of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions

Review the Decision Tree Task

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

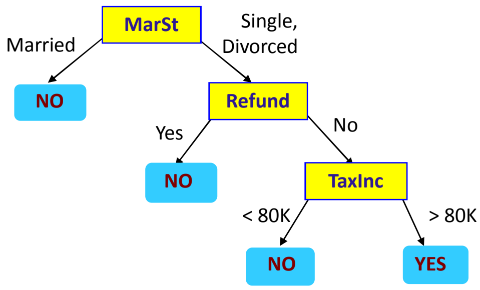


Model: Decision Tree

Review the Decision Tree Task

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!

15

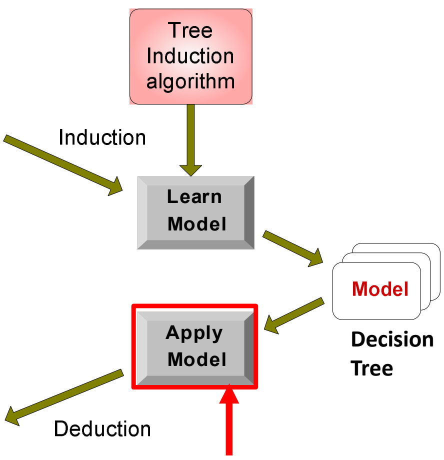
Review the Decision Tree Task

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	?

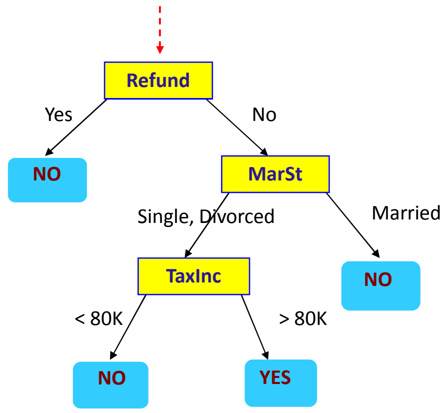
Test Set



16

Review the Decision Tree Task

Start from the root of tree.



Test Data

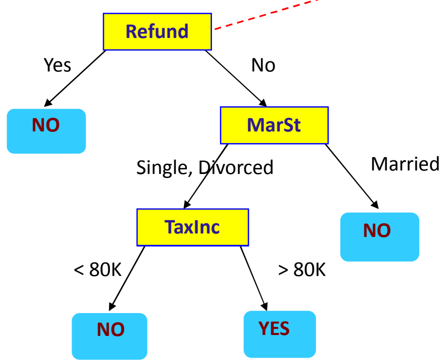
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

17

Review the Decision Tree Task

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

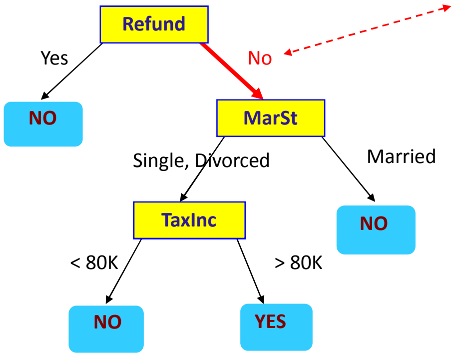


18

Review the Decision Tree Task

Test Data

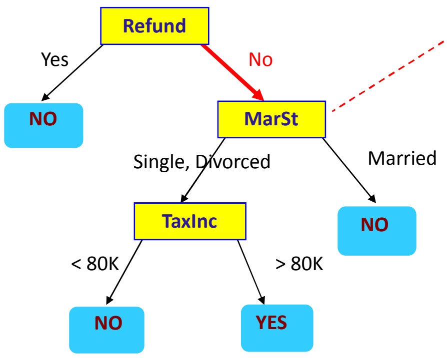
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Review the Decision Tree Task

Test Data

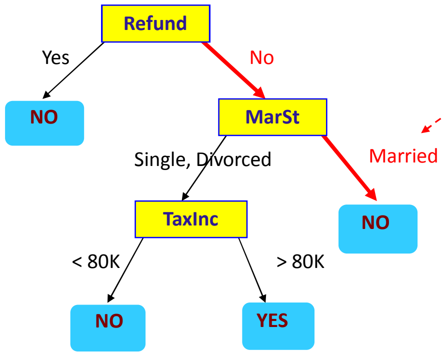
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Review the Decision Tree Task

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

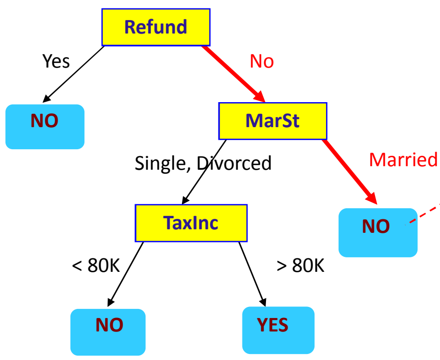


21

Review the Decision Tree Task

Test Data

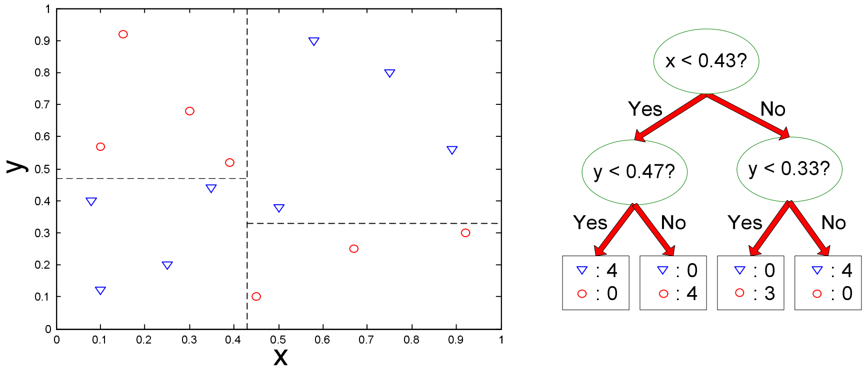
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

22

Decision Boundary



- Border line between two neighboring regions of different classes is known as decision boundary
- Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

Other Attribute Selection Measures

- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- C-SEP: performs better than info. gain and gini index in certain cases
- G-statistics: has a close approximation to χ^2 distribution
- MDL (Minimal Description Length) principle (i.e., the simplest solution is preferred):
  - The best tree as the one that requires the fewest number of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- Multivariate splits (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear comb. of attrs.
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

Random Forest (Breiman 2001)

- **Random Forest:**
  - Each classifier in the ensemble is a *decision tree* classifier and is generated using a random selection of attributes at each node to determine the split
  - During classification, each tree votes and the most popular class is returned
- **Two Methods to construct Random Forest**
  - Forest-RI (*random input selection*): Randomly select, at each node, F attributes as candidates for the split at the node. The CART methodology is used to grow the trees to maximum size
  - Forest-RC (*random linear combinations*): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)
- **Comparable in accuracy to Adaboost, but more robust to errors and outliers**
- **Insensitive to the number of attributes selected for consideration at each split, and faster than bagging or boosting**

Extracting Classification Rules from Trees

- **Represent the knowledge in the form of IF-THEN rules**
- **One rule is created for each path from the root to a leaf**
- **Each attribute-value pair along a path forms a conjunction**
- **The leaf node holds the class prediction**
- **Rules are easier for humans to understand**
- **Example**

IF *age* = "<=30" AND *student* = "*no*" THEN *buys_computer* = "*no*"
IF *age* = "<=30" AND *student* = "*yes*" THEN *buys_computer* = "*yes*"
IF *age* = "31…40" THEN *buys_computer* = "*yes*"
IF *age* = ">40" AND *credit_rating* = "*excellent*" THEN *buys_computer* = "*yes*"
IF *age* = "<=30" AND *credit_rating* = "*fair*" THEN *buys_computer* = "*no*"

Overfitting and Tree Pruning

- **Overfitting**: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- **Two approaches to avoid overfitting**
  - **Prepruning**: *Halt tree construction early*-do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - **Postpruning**: *Remove branches* from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

Enhancements to basic decision tree induction

- **Allow for continuous-valued attributes**
  - **Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals**
- **Handle missing attribute values**
  - **Assign the most common value of the attribute**
  - **Assign probability to each of the possible values**
- **Attribute construction**
  - **Create new attributes based on existing ones that are sparsely represented**
  - **This reduces fragmentation(碎片), repetition（重复）, and replication（复制）**

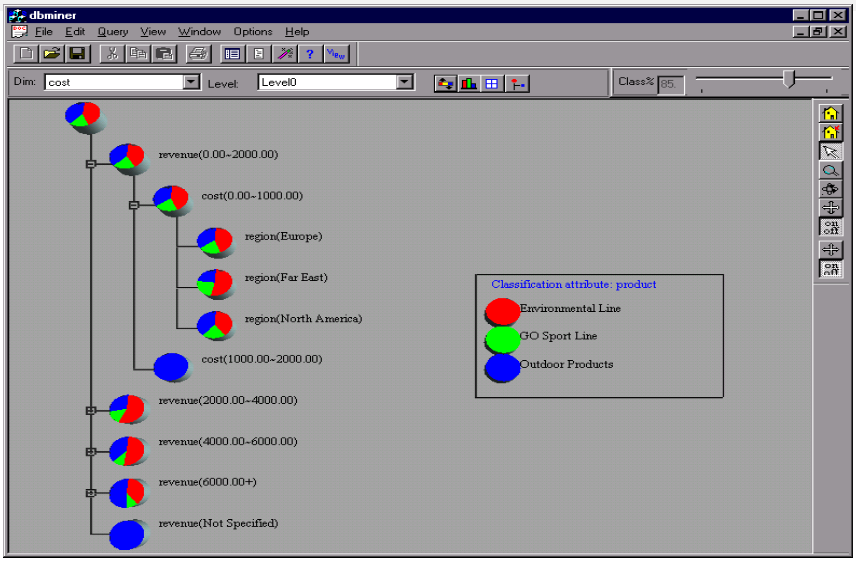
Classification in Large Databases

- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- Why decision tree induction in data mining?
  - relatively faster learning speed (than other classification methods)
  - convertible to simple and easy to understand classification rules
  - can use SQL queries for accessing databases
  - comparable classification accuracy with other methods

Scalable Decision Tree Induction Methods in Data Mining Studies

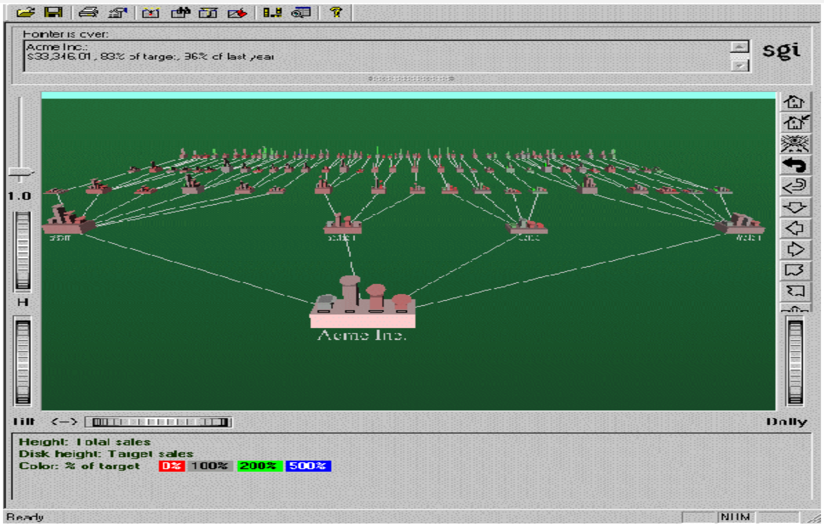
- SLIQ (EDBT'96 — Mehta et al.)
  - builds an index for each attribute and only class list and the current attribute list reside in memory
- SPRINT (VLDB'96 — J. Shafer et al.)
  - constructs an attribute list data structure
- PUBLIC (VLDB'98 — Rastogi & Shim)
  - integrates tree splitting and tree pruning: stop growing the tree earlier
- RainForest (VLDB'98 — Gehrke, Ramakrishnan & Ganti)
  - separates the scalability aspects from the criteria that determine the quality of the tree
  - builds an AVC-list (attribute, value, class label)

Presentation of Classification Results



31

Visualization of a Decision Tree in SGI/MineSet 3.0



32

Thanks !