
COMP9318: Data Warehousing and Data Mining

— L7: Classification and Prediction —

Modified from Prof. Jiawei Han's Slides

Chapter 7. Classification and Prediction

- **What is classification? What is prediction?**
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

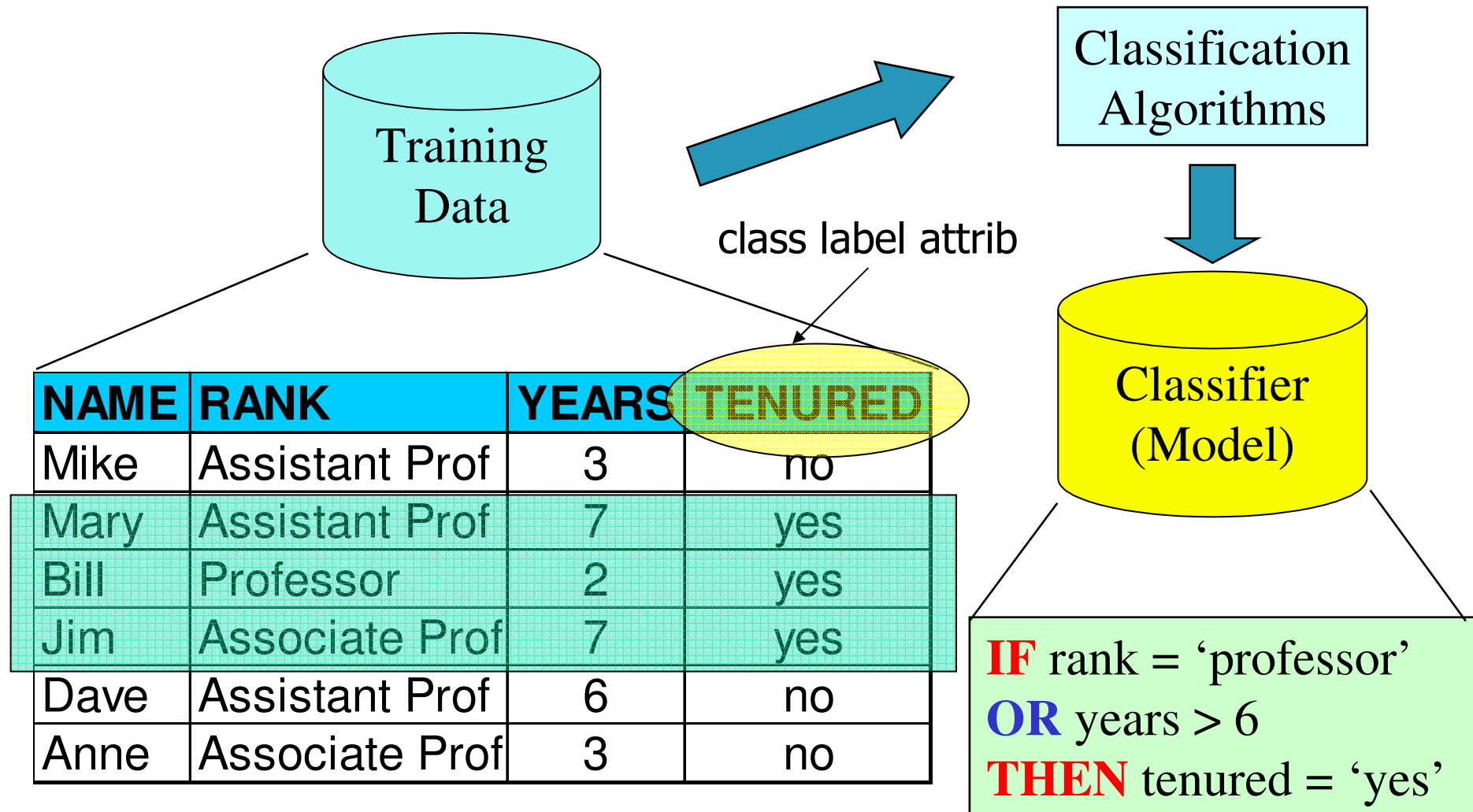
Classification vs. Prediction

- **Classification:**
 - predicts categorical class labels (discrete or nominal)
 - classifies data (constructs a model) based on the training set and the values (**class labels**) in a classifying attribute and uses it in classifying new data
- **Prediction:**
 - models continuous-valued functions, i.e., predicts unknown or missing values
- **Typical Applications**
 - credit approval
 - target marketing
 - medical diagnosis
 - treatment effectiveness analysis

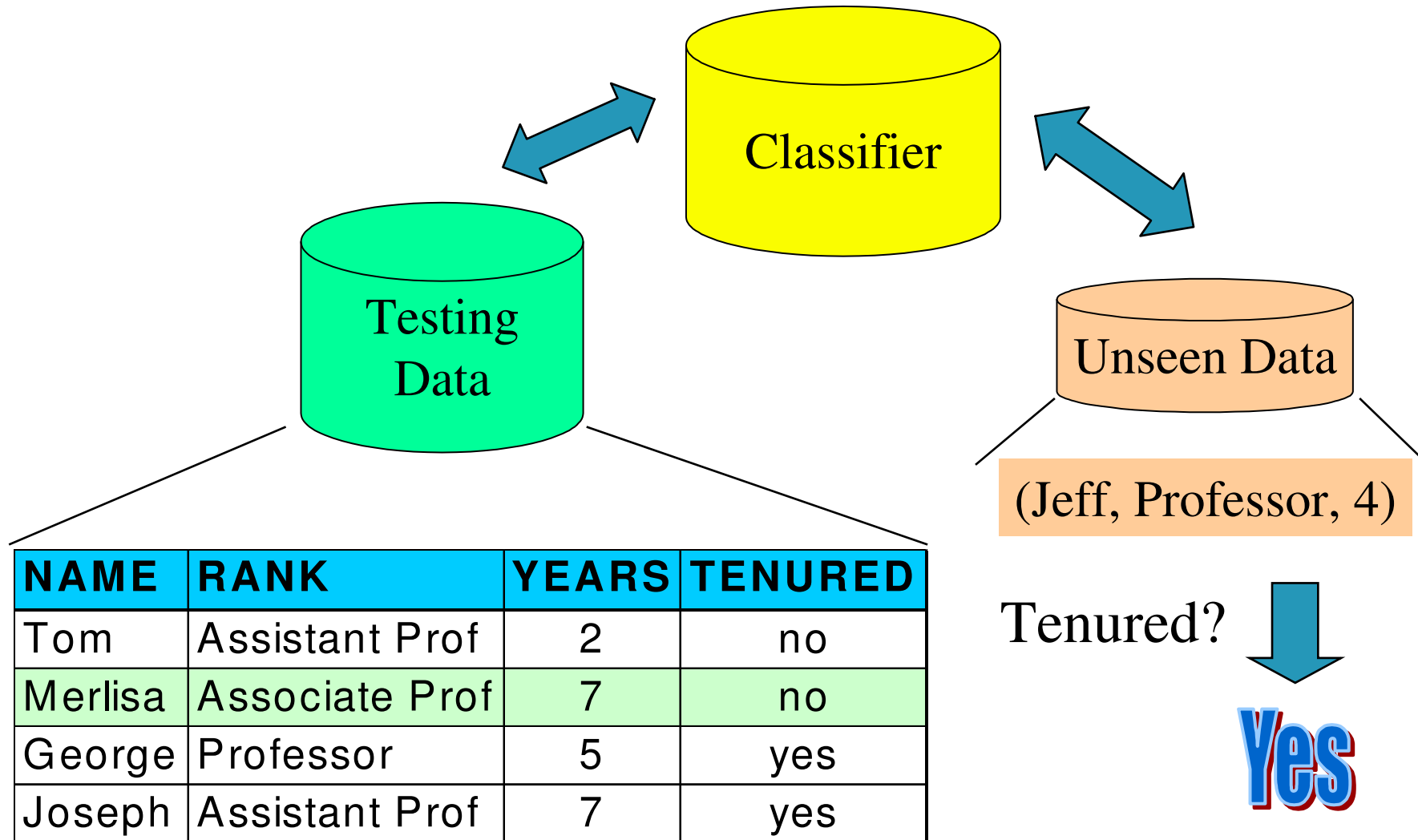
Classification—A Two-Step Process

- **Model construction**: describing a set of predetermined classes
 - Each tuple/sample is assumed to belong to a predefined class, as determined by the **class label attribute**
 - The set of tuples used for model construction is **training set**
 - The model is represented as classification rules, decision trees, or mathematical formulae
- **Model usage**: for classifying future or unknown objects
 - Estimate accuracy of the model
 - The known label of test sample is compared with the classified result from the model
 - Accuracy rate is the percentage of test set samples that are correctly classified by the model
 - Test set is independent of training set, otherwise over-fitting will occur
 - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Classification Process (1): Model Construction



Classification Process (2): Use the Model in Prediction



Supervised vs. Unsupervised Learning

- **Supervised learning (classification)**
 - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
 - New data is classified based on the training set
- **Unsupervised learning (clustering)**
 - The class labels of training data is unknown
 - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- **Issues regarding classification and prediction**
- Classification by decision tree induction
- Bayesian Classification
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Issues Regarding Classification and Prediction (1): Data Preparation

- Data cleaning
 - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
 - Remove the irrelevant or redundant attributes
- Data transformation
 - Generalize and/or normalize data

Issues regarding classification and prediction (2): Evaluating Classification Methods

- Predictive accuracy
- Speed and scalability
 - time to construct the model
 - time to use the model
- Robustness
 - handling noise and missing values
- Scalability
 - efficiency in disk-resident databases
- Interpretability:
 - understanding and insight provided by the model
- Goodness of rules
 - decision tree size
 - compactness of classification rules

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- **Classification by decision tree induction**
- Bayesian Classification
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Training Dataset

This follows an example from Quinlan's ID3

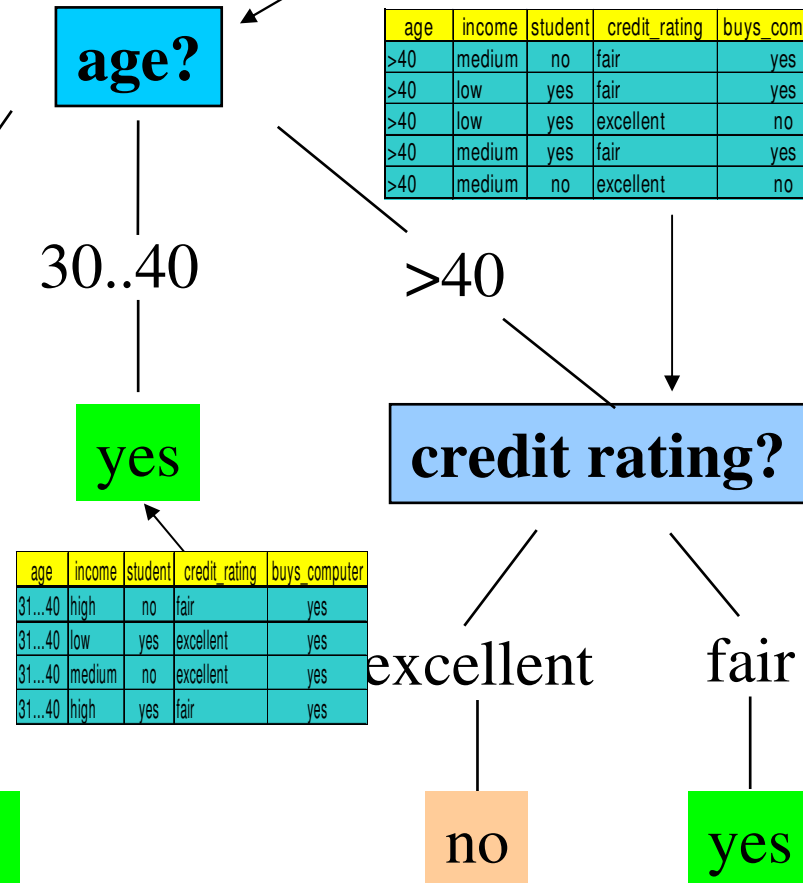
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes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

Output: A Decision Tree for

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

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

age	income	student	credit_rating	buys_computer
>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



age	income	student	credit_rating	buys_computer
31...40	high	no	fair	yes
31...40	low	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes

Algorithm for Decision Tree Induction

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

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

- Select the attribute with the **highest information gain**
- S contains s_i tuples of class C_i for $i = \{1, \dots, m\}$
- **information** measures info required to classify any arbitrary tuple

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

- **entropy** of attribute A with values $\{a_1, a_2, \dots, a_v\}$

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

- **information gained** by branching on attribute A

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

Attribute Selection by Information Gain Computation

- 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

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

$\frac{5}{14} I(2,3)$ means "age ≤ 30 " has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence
 $Gain(\text{age}) = I(p, n) - E(\text{age}) = 0.246$

Similarly,

$$Gain(\text{income}) = 0.029$$

$$Gain(\text{student}) = 0.151$$

$$Gain(\text{credit_rating}) = 0.048$$

income	p_i	n_i	$I(p_i, n_i)$
high	2	2	1
medium	4	2	0.918

Q: what's the extreme/worst case?

Other Attribute Selection Measures

- **Gini index** (CART, IBM IntelligentMiner)
 - All attributes are assumed continuous-valued
 - Assume there exist several possible split values for each attribute
 - May need other tools, such as clustering, to get the possible split values
 - Can be modified for categorical attributes

Gini Index (IBM IntelligentMiner)

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

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

where p_j is the **relative** frequency of class j in T .

- If a data set T is split into two subsets T_1 and T_2 with sizes N_1 and N_2 respectively, the $gini$ index of the split data contains examples from n classes, the $gini$ index $gini_{split}(T)$ is defined as

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- The attribute provides the **smallest** $gini_{split}(T)$ is chosen to split the node (*need to enumerate all possible splitting points for each attribute*).

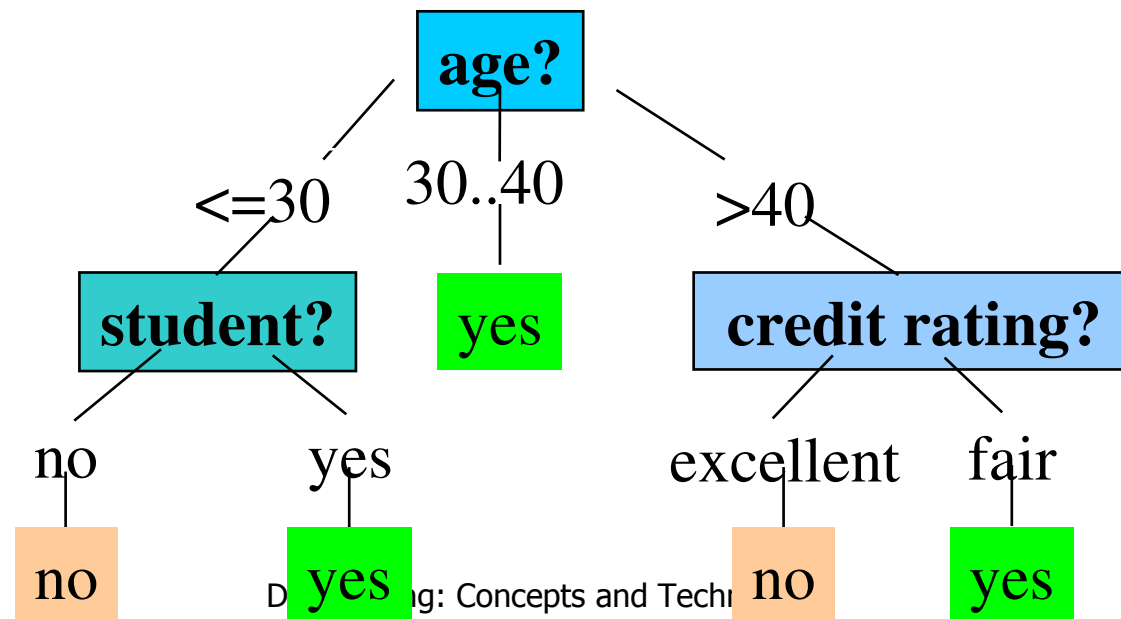
Extracting Classification Rules from Trees

- Represent the knowledge in the form of **IF-THEN** rules
 - Rules are easier for humans to understand
- 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
 - Example

IF *age* = " ≤ 30 " **AND** *student* = "*no*" **THEN** *buys_computer* = "*no*"

IF *age* = " ≤ 30 " **AND** *student* = "*yes*" **THEN** *buys_computer* = "*yes*"

... ..



Avoid Overfitting in Classification

- **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”

Approaches to Determine the Final Tree Size

- Use a separate validation set
- Use all the data for training
 - but apply a **statistical test** (e.g., chi-square) to estimate whether expanding or pruning a node may improve the entire distribution
- Use minimum description length (MDL) principle
 - halting growth of the tree when the encoding is minimized

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, repetition, and replication

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)
- Related: **GAC** (VLDB'00 — Lu & Liu)
 - build “decision table” by SQL with OLAP extensions

SPRINT

Shafer, Agrawal, Mehta (VLDB 1996)

- Motivation:
 - Scalable data access method for CART
 - Improvement over SLIQ to avoid main-memory data structure
- Ideas:
 - Create vertical partitions called **attribute lists** for each attribute
 - Pre-sort the attribute lists

Recursive tree construction:

1. Scan all attribute lists at node t to find the best split
2. Partition current attribute lists over children nodes while maintaining sort orders
3. *Call SPRINT on each partition recursively*

SPRINT: Attribute Lists

sorted for
continuous attributes

	Age	Car	Class
1	20	M	Y
2	30	M	Y
3	25	T	N
4	30	S	Y
5	40	S	Y
6	20	T	N
7	30	M	Y
8	25	M	Y
9	40	M	Y
10	20	S	N

data

Age	Class	Ind
20	Y	1
20	N	6
20	N	10
25	N	3
25	Y	8
30	Y	2
30	Y	4
30	Y	7
40	Y	5
40	Y	9

attrib list for **Age**

attrib list for **Car**

Car	Class	Ind
M	Y	1
M	Y	2
T	N	3
S	Y	4
S	Y	5
T	N	6
M	Y	7
M	Y	8
M	Y	9
S	N	10

otherwise, just a

vertical partitioning

SPRINT: Evaluation of Splits

attrib list for **Age**

Age	Class	Ind
20	Y	1
20	N	6
20	N	10
25	N	3
25	Y	8
30	Y	2
30	Y	4
30	Y	7
40	Y	5
40	Y	9

Scan

← pos=0

← pos=3

pos=0	Y	N
below	7	3
above	0	0

$$Gini_{split}(S) = 10/10 * Gini(7,3) + 0/10 * Gini(0,0) = 0.42$$

pos=3	Y	N
below	6	1
above	1	2

$$Gini_{split}(S) = 3/10 * Gini(1,2) + 7/10 * Gini(6,1) = 0.30$$

...

...

...

$$Gini(S) = 1 - \sum p_j^2$$

$$Gini_{split}(S) = \frac{n_1}{n} Gini(S_1) + \frac{n_2}{n} Gini(S_2)$$

Case I: numeric attribs

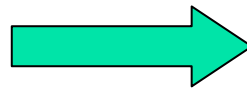
Exercise: compute gini indexes for all possible splits

SPRINT: Evaluation of Splits

count matrix

attrib list for **Car**

Car	Class	Ind
M	Y	1
M	Y	2
T	N	3
S	Y	4
S	Y	5
T	N	6
M	Y	7
M	Y	8
M	Y	9
S	N	10



	Class=Y	Class=N
M	5	0
T	0	2
S	2	1

Need to consider all possible splits !

	Y	N
{M, T}	5	2
{S}	2	1

	Y	N
{M, S}	7	1
{T}	0	2

	Y	N
{T, S}	2	3
{M}	5	0

$$\begin{aligned} \text{Gini}_{\text{split}}(S) &= \\ &= \frac{7}{10} * \text{Gini}(5,2) + \\ &= \frac{3}{10} * \text{Gini}(2,1) = \\ &= 0.42 \end{aligned}$$

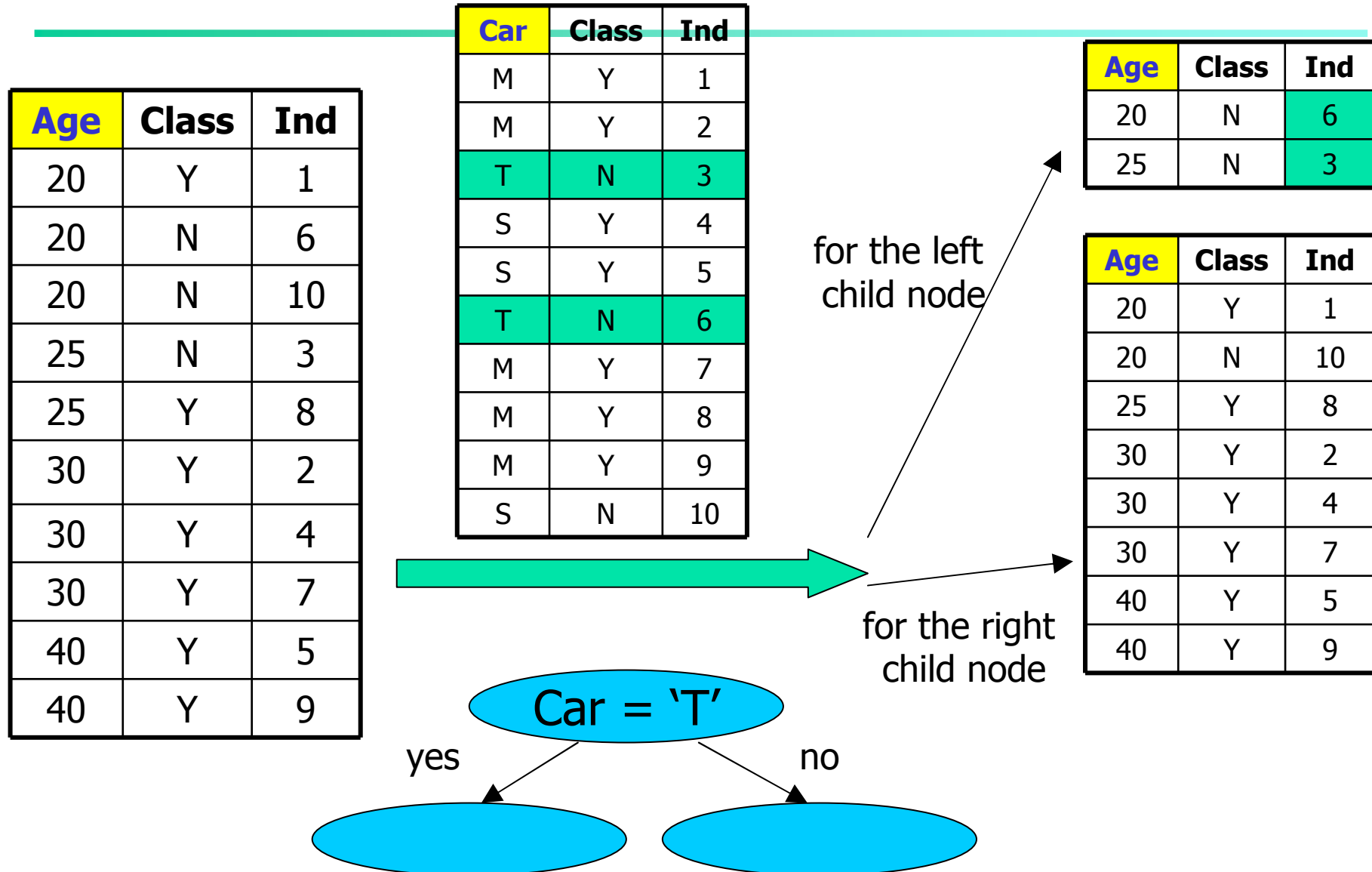
$$\begin{aligned} \text{Gini}_{\text{split}}(S) &= \\ &= \frac{8}{10} * \text{Gini}(7,1) + \\ &= \frac{2}{10} * \text{Gini}(0,2) = \\ &= \mathbf{0.18} \end{aligned}$$

$$\begin{aligned} \text{Gini}_{\text{split}}(S) &= \\ &= \frac{5}{10} * \text{Gini}(2,3) + \\ &= \frac{5}{10} * \text{Gini}(5,0) = \\ &= 0.24 \end{aligned}$$

SPRINT: Splitting of a Node

1. Scan all attribute lists to find the best split
2. Partition the attribute list of the splitting attribute X
3. For each attribute $X_i \neq X$
 - Perform the partitioning step of a hash-join between the attribute list of X and the attribute list of X_i

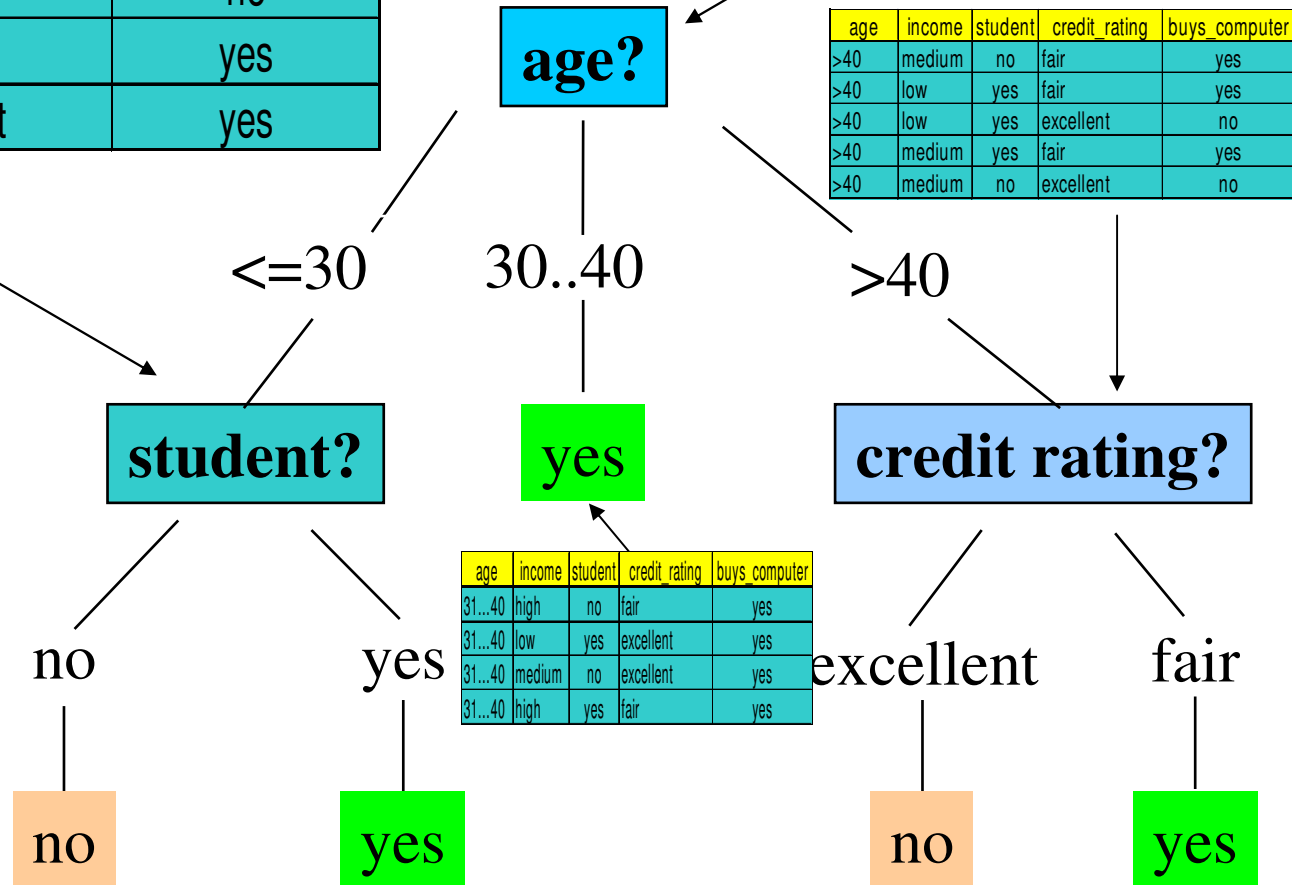
SPRINT: Hash-Join Partitioning



ID3

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

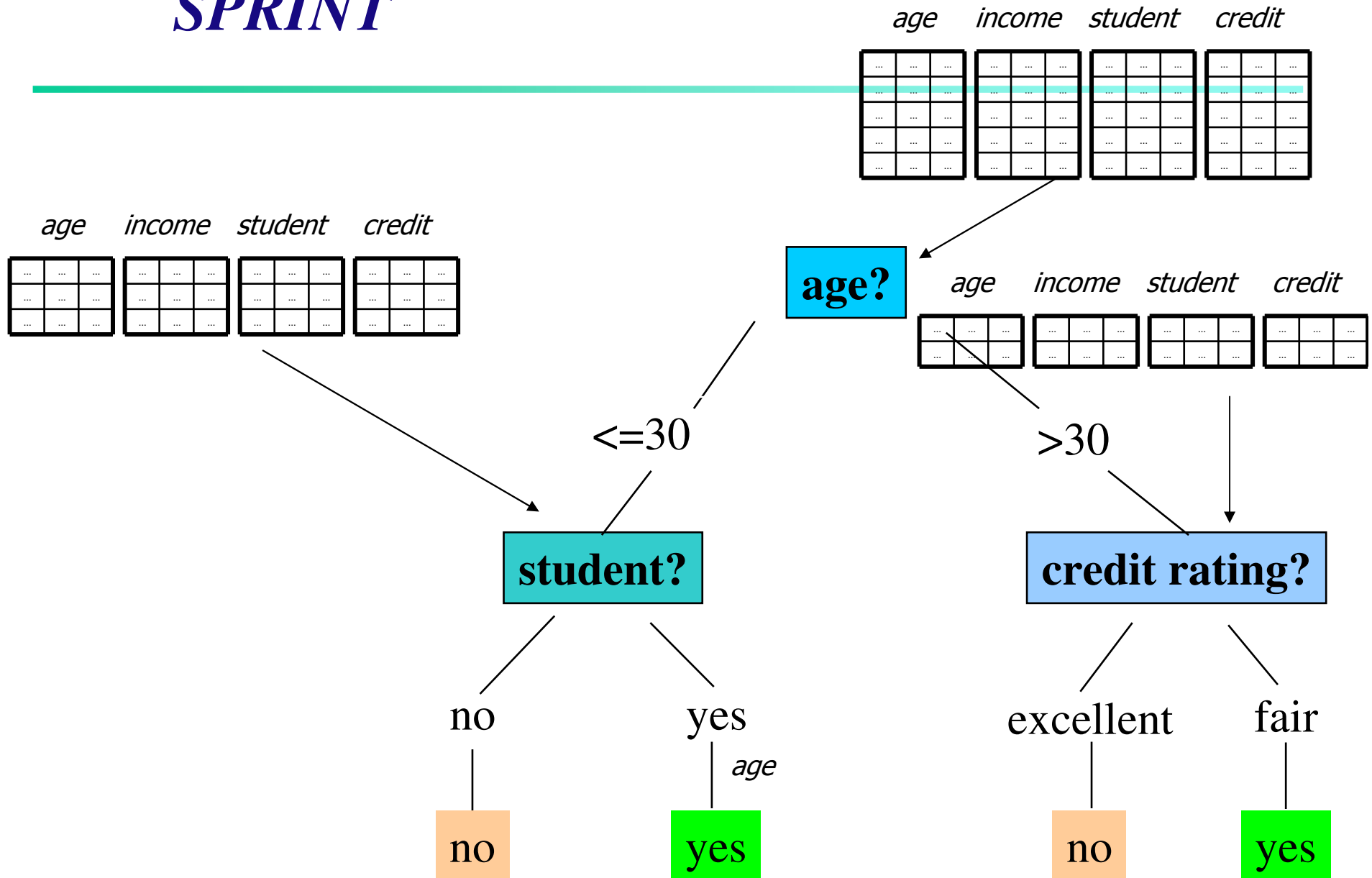
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes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



age	income	student	credit_rating	buys_computer
>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

age	income	student	credit_rating	buys_computer
31...40	high	no	fair	yes
31...40	low	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes

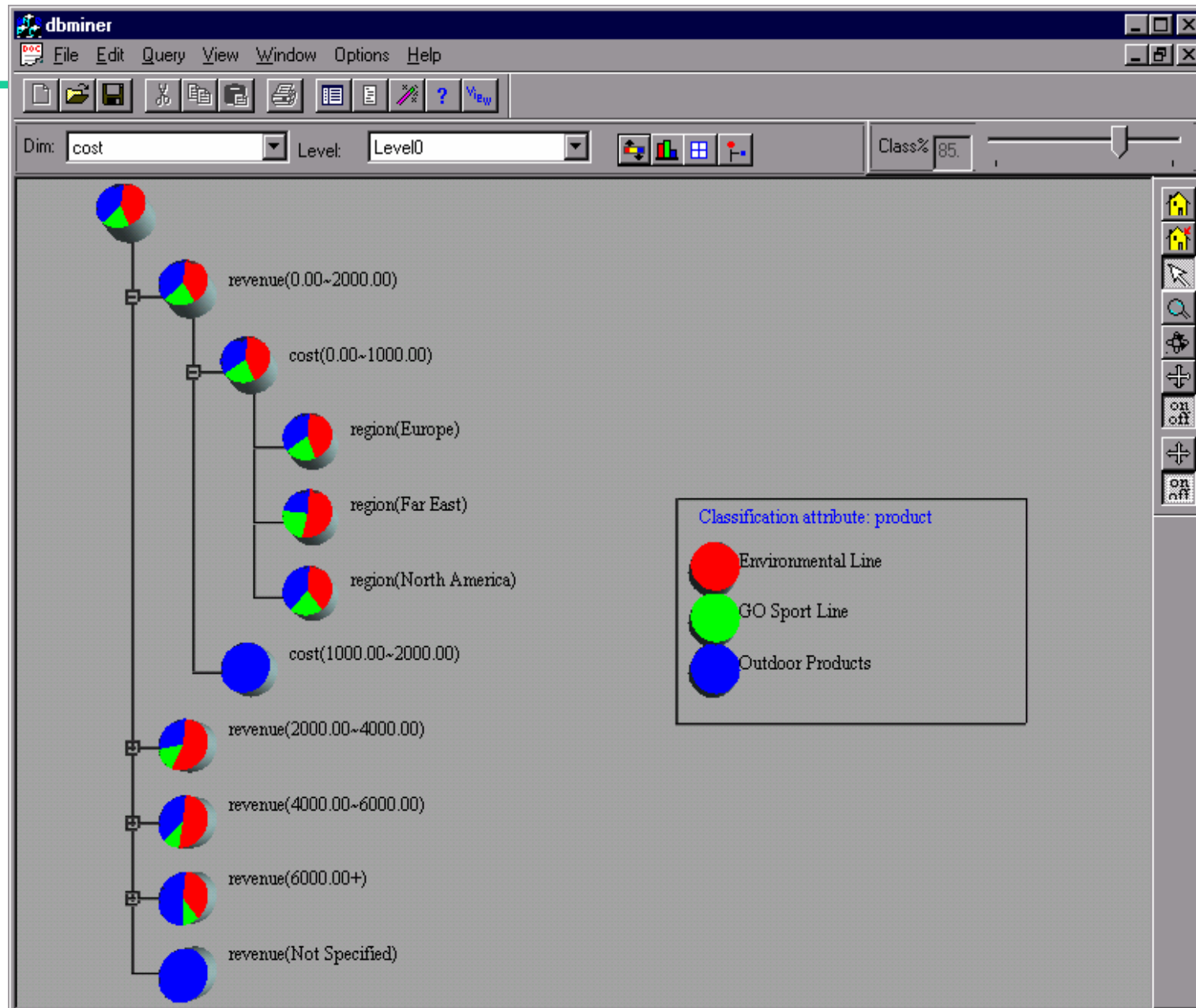
SPRINT



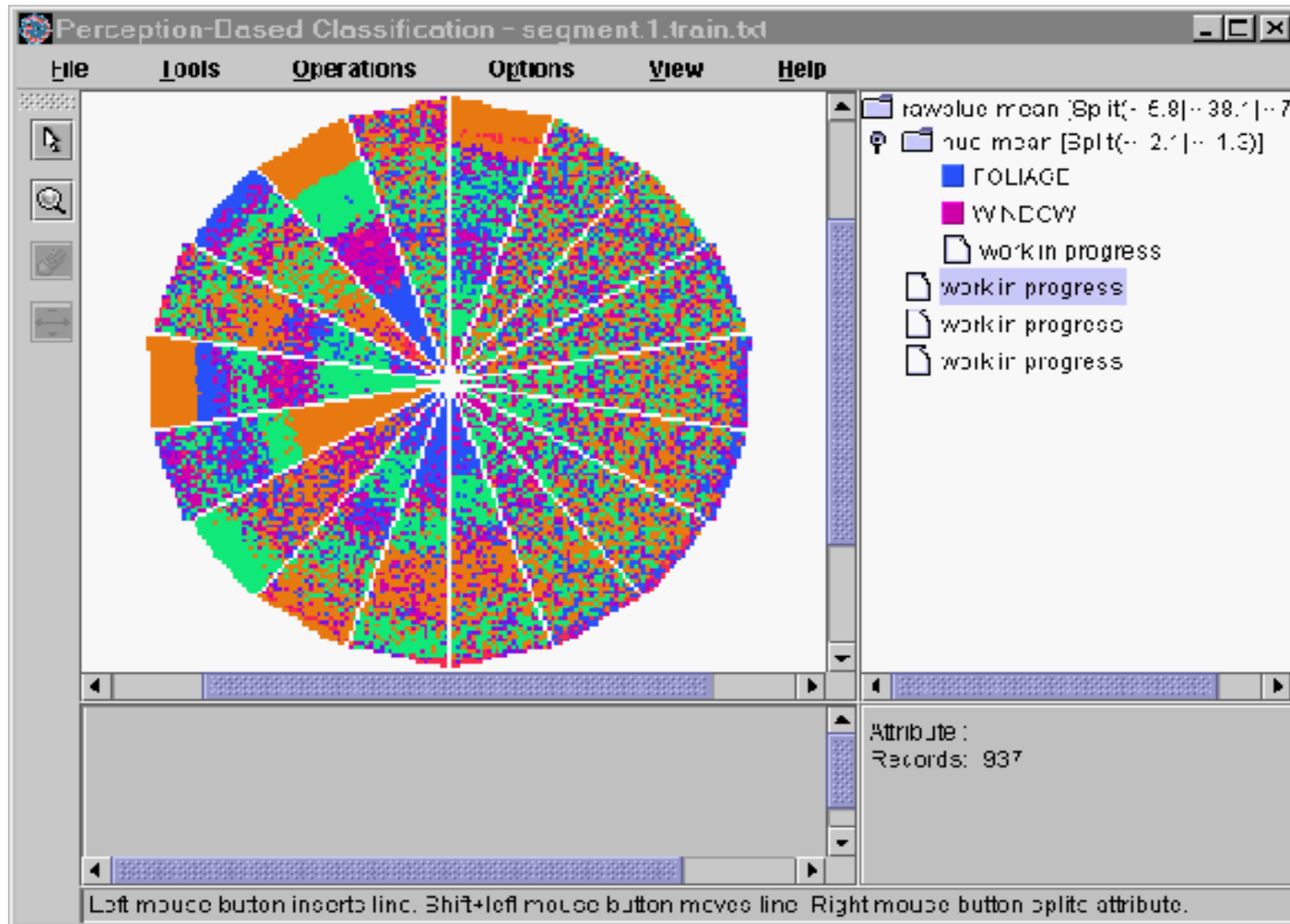
SPRINT: Summary

- Scalable data access method for CART split selection method
- Completely scalable, can be (and has been) implemented “inside” a database system
- Hash-join partitioning step expensive (each attribute, at each node of the tree)

Presentation of Classification Results



Interactive Visual Mining by Perception-Based Classification (PBC)



Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- **Bayesian Classification**
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Bayesian Classification: Why?

- Probabilistic learning: Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.
- Probabilistic prediction: **Predict multiple hypotheses**, weighted by their probabilities
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayesian Theorem: Basics

- Let X be a data sample whose class label is **unknown**
- Let H be a **hypothesis** that X belongs to class C
- For classification problems, determine $P(H|X)$: the probability that the hypothesis holds given the observed data sample X
- $P(H)$: **prior** probability of hypothesis H (i.e. the initial probability before we observe any data, reflects the background knowledge)
- $P(X)$: probability that sample data is observed
- $P(X|H)$: probability of observing the sample X , given that the hypothesis holds

Bayesian Theorem

- Given training data X , *posteriori probability of a hypothesis* H , $P(H|X)$ follows the Bayes theorem

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

P(java | coffee)
vs. P(java | lang)

- Informally, this can be written as
posterior = likelihood x prior / evidence
- MAP (maximum posteriori) hypothesis

$$h_{MAP} \equiv \arg \max_{h \in H} P(h|D) = \arg \max_{h \in H} P(D|h)P(h).$$

- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Naïve Bayes Classifier

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

- A simplified assumption: attributes are conditionally independent:

$$P(X | C_i) = \prod_{k=1}^n P(x_k | C_i)$$

P(java class | coffee)
vs. P(java class | lang)

- The product of occurrence of say 2 elements x_1 and x_2 , given the current class is C , is the product of the probabilities of each element taken separately, given the same class $P([x_1, x_2], C) = P(x_1, C) * P(x_2, C)$
- No dependence relation between attributes
- Greatly reduces the computation cost, only count the class distribution.
- Once the probability $P(X|C_i)$ is known, assign X to the class with maximum $P(X|C_i)*P(C_i)$

Training dataset

Class:

C1:buys_computer=
'yes'

C2:buys_computer=
'no'

Data sample

X =(age<=30,
Income=medium,
Student=yes,
Credit_rating=
Fair)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30...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

Exercise: Google translation: "java", "java trip", "java class trip"
java, 爪哇之旅, Java类访问

Naïve Bayesian Classifier: Example

- Compute $P(X|C_i)$ for each class

$X=(age \leq 30, income = medium, student = yes, credit_rating = fair)$

$$P(age = "<30" | buys_computer = "yes") = 2/9 = 0.222$$

$$P(age = "<30" | buys_computer = "no") = 3/5 = 0.6$$

$$P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444$$

$$P(income = "medium" | buys_computer = "no") = 2/5 = 0.4$$

$$P(student = "yes" | buys_computer = "yes") = 6/9 = 0.667$$

$$P(student = "yes" | buys_computer = "no") = 1/5 = 0.2$$

$$P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667$$

$$P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4$$

$$P(X | C_i) : P(X | buys_computer = "yes") = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$

$$P(X | buys_computer = "no") = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

$$P(X | C_i) * P(C_i) : P(X | buys_computer = "yes") * P(buys_computer = "yes") = 0.028$$

$$P(X | buys_computer = "no") * P(buys_computer = "no") = 0.007$$

X belongs to class "buys_computer=yes"

likelihood

Naïve Bayesian Classifier: Comments

- Advantages :
 - Easy to implement
 - Good results obtained in most of the cases
- Disadvantages
 - Assumption: class conditional independence , therefore loss of accuracy
 - Practically, dependencies exist among variables
 - E.g., hospitals: patients: Profile: age, family history etc
Symptoms: fever, cough etc., Disease: lung cancer, diabetes etc
 - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
 - **Bayesian Belief Networks**

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Association-Based Classification

- Several methods for association-based classification
 - ARCS: Quantitative association mining and clustering of association rules (Lent et al'97)
 - It beats C4.5 in (mainly) scalability and also accuracy
 - Associative classification: (Liu et al'98)
 - It mines high support and high confidence rules in the form of "cond_set => y", where **y is a class label**
 - CAEP (Classification by aggregating emerging patterns) (Dong et al'99)
 - Emerging patterns (EPs): the itemsets whose support increases significantly from one class to another
 - Mine Eps based on minimum support and growth rate

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification based on concepts from association rule mining
- **Other Classification Methods**
- Prediction
- Classification accuracy
- Summary

Other Classification Methods

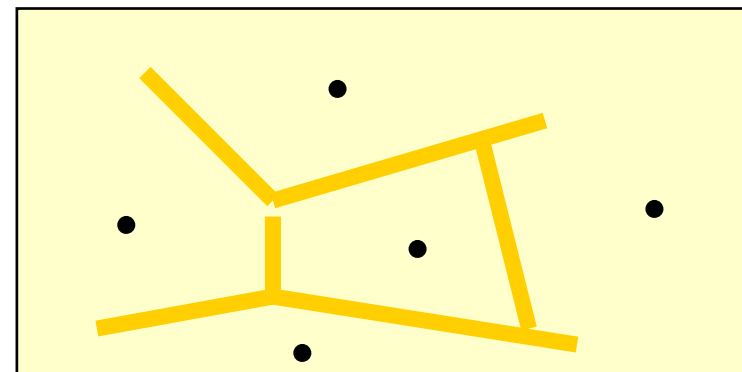
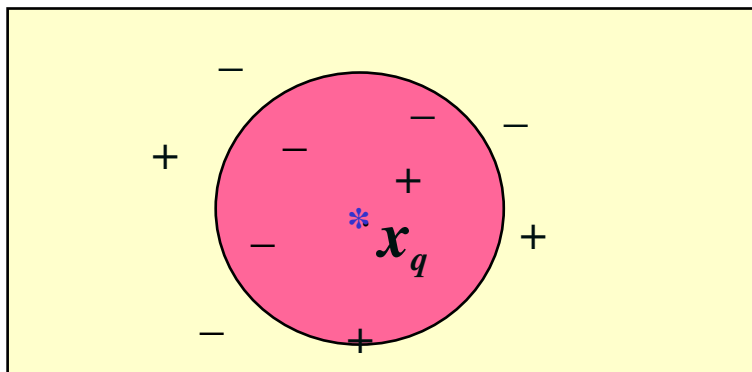
- k-nearest neighbor classifier
- case-based reasoning
- Genetic algorithm
- Rough set approach
- Fuzzy set approaches

Instance-Based Methods

- Instance-based learning:
 - Store training examples and delay the processing (“lazy evaluation”) until a new instance must be classified
- Typical approaches
 - *k*-nearest neighbor approach
 - Instances represented as points in a Euclidean space.
 - Locally weighted regression
 - Constructs local approximation
 - Case-based reasoning
 - Uses symbolic representations and knowledge-based inference

The k -Nearest Neighbor Algorithm

- All instances correspond to points in the n -D space.
- The nearest neighbor are defined in terms of Euclidean distance.
- The target function could be discrete- or real- valued.
- For discrete-valued, the k -NN returns the most common value among the k training examples nearest to x_q .
- Voronoi diagram: the decision surface induced by 1-NN for a typical set of training examples.



Discussion on the k -NN Algorithm

- The k -NN algorithm for continuous-valued target functions
 - Calculate the mean values of the k nearest neighbors
- **Distance-weighted** nearest neighbor algorithm
 - Weight the contribution of each of the k neighbors according to their distance to the query point x_q
 - giving greater weight to closer neighbors
 - Similarly, for real-valued target functions
- Robust to noisy data by averaging k -nearest neighbors
- Curse of dimensionality:
 - k NN search becomes very expensive in high dimensional space
 - distance between neighbors could be dominated by irrelevant attributes.
 - To overcome it, axes stretch or elimination of the least relevant attributes.

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

Remarks on Lazy vs. Eager Learning

- Instance-based learning: lazy evaluation
- Decision-tree and Bayesian classification: eager evaluation
- Key differences
 - Lazy method may consider query instance x_q when deciding how to generalize beyond the training data D
 - Eager method cannot since they have already chosen global approximation when seeing the query
- Efficiency: Lazy - less time training but more time predicting
- Accuracy
 - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
 - Eager: must commit to a single hypothesis that covers the entire instance space

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

What Is Prediction?

- Prediction is similar to classification
 - First, construct a model
 - Second, use model to predict unknown value
 - Major method for prediction is regression
 - Linear and multiple regression
 - Non-linear regression
- Prediction is different from classification
 - Classification refers to predict categorical class label
 - Prediction models continuous-valued functions

Predictive Modeling in Databases

- Predictive modeling: Predict data values or construct generalized linear models based on the database data.
- One can only predict value ranges or category distributions
- Method outline:
 - Minimal generalization
 - Attribute relevance analysis
 - Generalized linear model construction
 - Prediction
- Determine the major factors which influence the prediction
 - Data relevance analysis: uncertainty measurement, entropy analysis, expert judgement, etc.
- Multi-level prediction: drill-down and roll-up analysis

Regress Analysis and Log-Linear Models in Prediction

- Linear regression: $Y = \alpha + \beta X$
 - Two parameters , α and β specify the line and are to be estimated by using the data at hand.
 - using the least squares criterion to the known values of $Y_1, Y_2, \dots, X_1, X_2, \dots$
- Multiple regression: $Y = b_0 + b_1 X_1 + b_2 X_2$.
 - Many nonlinear functions can be transformed into the above.
- Log-linear models:
 - The multi-way table of joint probabilities is approximated by a product of lower-order tables.
 - Probability: $p(a, b, c, d) = \alpha_{ab} + \beta_{ac} + \chi_{ad} + \delta_{bcd}$

Locally Weighted Regression

- Construct an explicit approximation to f over a local region surrounding query instance x_q .
- Locally weighted linear regression:

- The target function f is approximated near x_q using the linear function:
- minimize the squared error: distance-decreasing weight

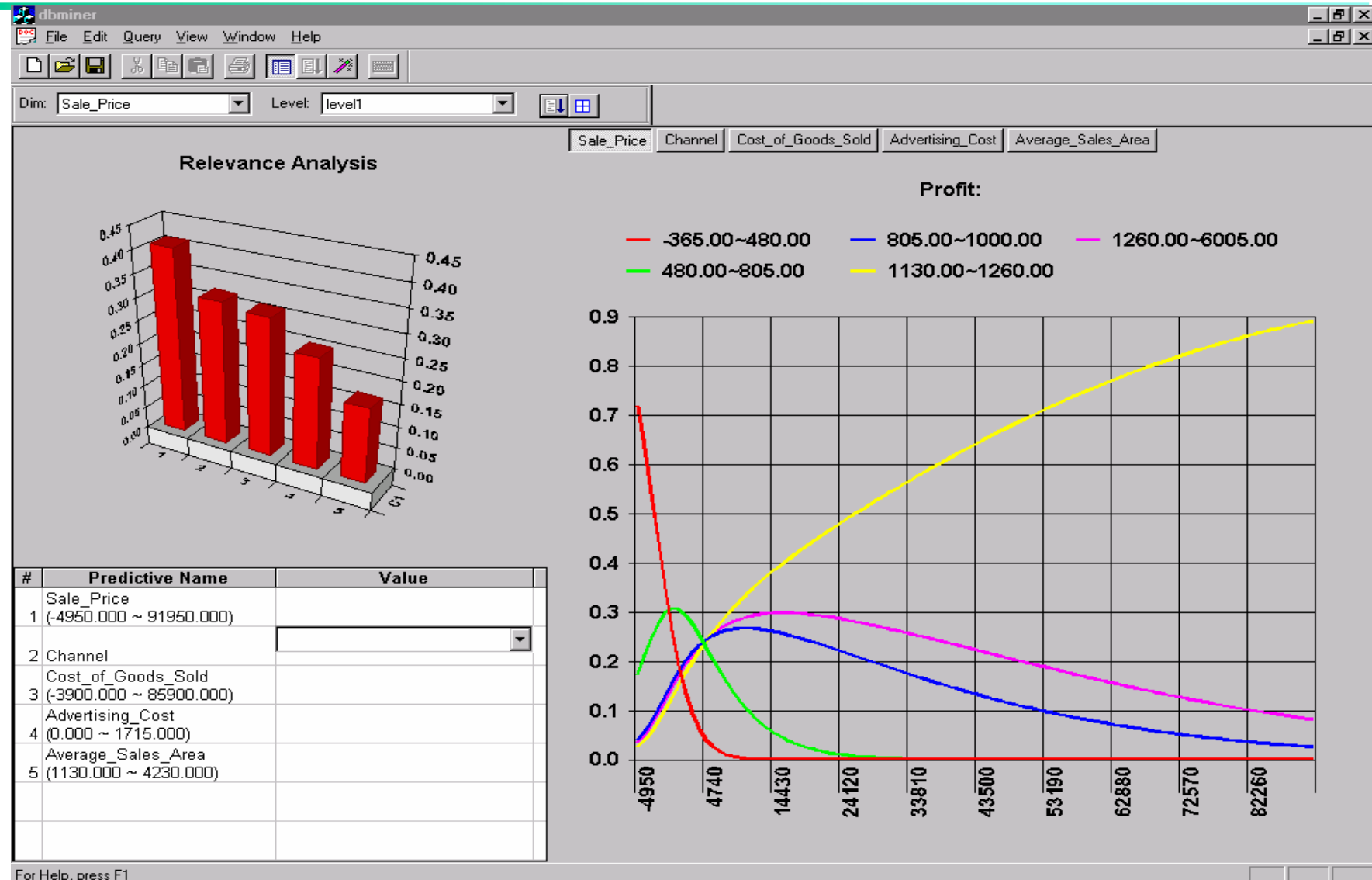
$$E(x_q) \equiv \frac{1}{2} \sum_{x \in kNN(x_q)} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$$

- the gradient descent training rule:

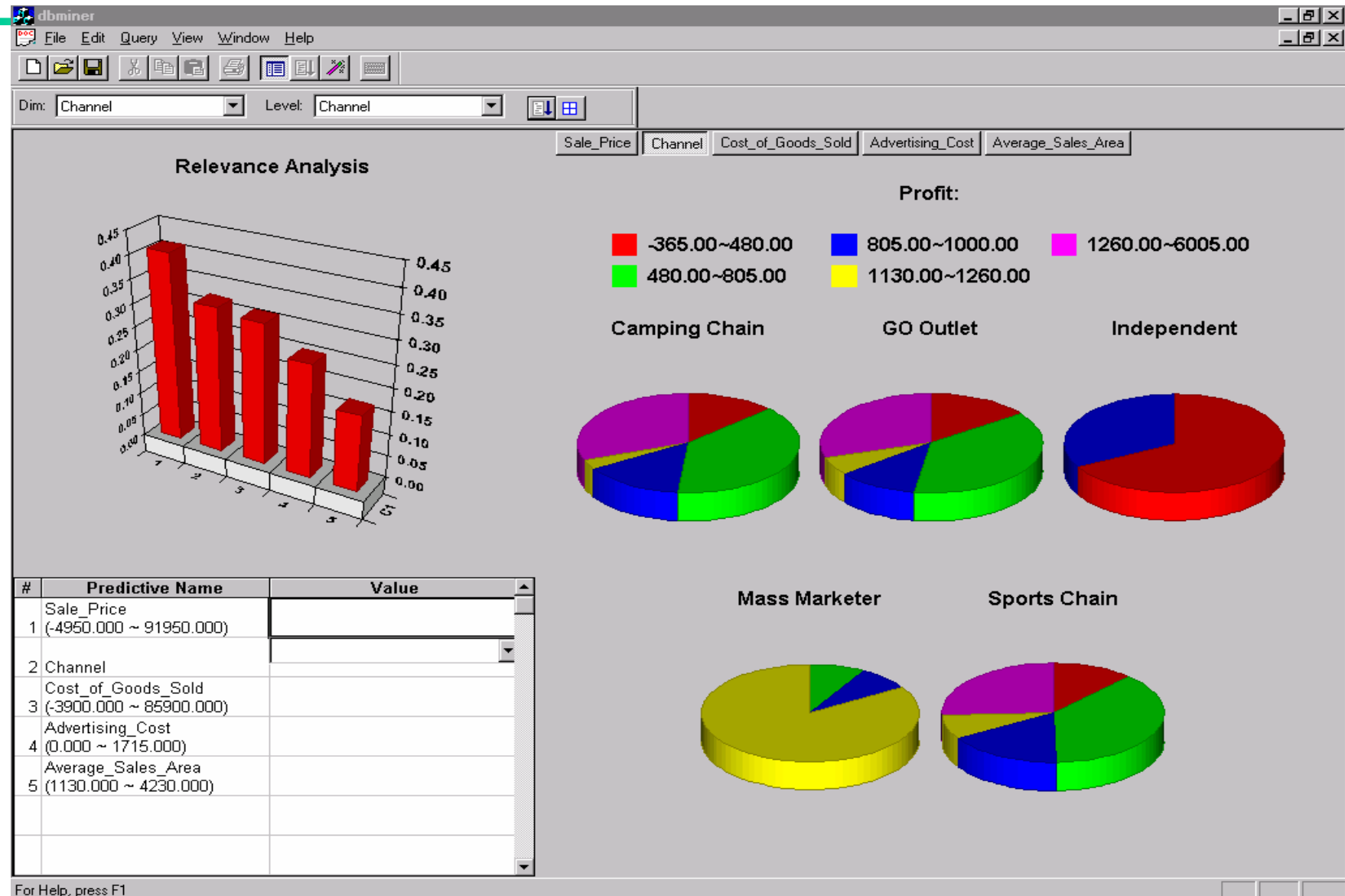
$$\Delta w_j \equiv \eta \sum_{x \in kNN(x_q)} K(d(x_q, x)) ((f(x) - \hat{f}(x)) a_j(x))$$

- In most cases, the target function is approximated by a constant, linear, or quadratic function.

Prediction: Numerical Data



Prediction: Categorical Data



Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- **Classification accuracy**
- Summary

Classification Accuracy: Estimating Error Rates

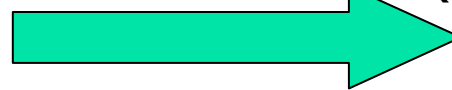
- Partition: Training-and-testing
 - use two independent data sets, e.g., training set (2/3), test set(1/3)
 - used for data set with large number of samples
- Cross-validation
 - divide the data set into k subsamples
 - use $k-1$ subsamples as training data and one subsample as test data— k -fold cross-validation
 - for data set with moderate size
- Bootstrapping (leave-one-out)
 - for small size data

Bagging and Boosting

- General idea

Training data

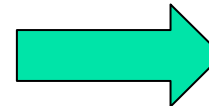
Classification method (CM)



Classifier C

Altered Training data

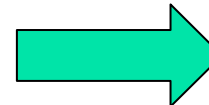
CM



Classifier C1

Altered Training data

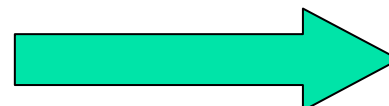
CM



Classifier C2

.....

Aggregation



Classifier C*

Bagging

- Given a set S of s samples
- Generate a bootstrap sample T from S . Cases in S may not appear in T or may appear more than once.
- Repeat this sampling procedure, getting a sequence of k independent training sets
- A corresponding sequence of classifiers C_1, C_2, \dots, C_k is constructed for each of these training sets, by using the same classification algorithm
- To classify an unknown sample X , let each classifier predict or vote
- The Bagged Classifier C^* counts the votes and assigns X to the class with the “most” votes

Boosting Technique — Algorithm

- Assign every example an equal weight $1/N$
- *For $t = 1, 2, \dots, T$ Do*
 - Obtain a hypothesis (classifier) $h^{(t)}$ under $w^{(t)}$
 - Calculate the error of $h^{(t)}$ and re-weight the examples based on the error. Each classifier is dependent on the previous ones. Samples that are incorrectly predicted are weighted more heavily
 - Normalize $w^{(t+1)}$ to sum to 1 (weights assigned to different classifiers sum to 1)
- Output a weighted sum of all the hypothesis, with each hypothesis weighted according to its accuracy on the training set

Bagging and Boosting

- Experiments with a new boosting algorithm, Freund et al (AdaBoost)
- Bagging Predictors, Breiman
- Boosting Naïve Bayesian Learning on large subset of MEDLINE, W. Wilbur

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- **Summary**

Summary

- Classification is an **extensively studied** problem (mainly in statistics, machine learning & neural networks)
- Classification is probably one of the most **widely used** data mining techniques with a lot of extensions
- **Scalability** is still an important issue for database applications: thus combining classification **with database techniques** should be a promising topic
- Research directions: classification of **non-relational data**, e.g., text, spatial, multimedia, etc..

References (1)

- C. Apte and S. Weiss. Data mining with decision trees and decision rules. *Future Generation Computer Systems*, 13, 1997.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.
- C. J. C. Burges. [A Tutorial on Support Vector Machines for Pattern Recognition](#). *Data Mining and Knowledge Discovery*, 2(2): 121-168, 1998.
- P. K. Chan and S. J. Stolfo. Learning arbiter and combiner trees from partitioned data for scaling machine learning. In Proc. 1st Int. Conf. Knowledge Discovery and Data Mining (KDD'95), pages 39-44, Montreal, Canada, August 1995.
- U. M. Fayyad. Branching on attribute values in decision tree generation. In Proc. 1994 AAAI Conf., pages 601-606, AAAI Press, 1994.
- J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest: A framework for fast decision tree construction of large datasets. In Proc. 1998 Int. Conf. Very Large Data Bases, pages 416-427, New York, NY, August 1998.
- J. Gehrke, V. Gant, R. Ramakrishnan, and W.-Y. Loh, [BOAT -- Optimistic Decision Tree Construction](#). In *SIGMOD'99*, Philadelphia, Pennsylvania, 1999

References (2)

- M. Kamber, L. Winstone, W. Gong, S. Cheng, and J. Han. Generalization and decision tree induction: Efficient classification in data mining. In Proc. 1997 Int. Workshop Research Issues on Data Engineering (RIDE'97), Birmingham, England, April 1997.
- B. Liu, W. Hsu, and Y. Ma. Integrating Classification and Association Rule Mining. *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98)* New York, NY, Aug. 1998.
- W. Li, J. Han, and J. Pei, CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules, Proc. 2001 Int. Conf. on Data Mining (ICDM'01), San Jose, CA, Nov. 2001.
- J. Magidson. The Chaid approach to segmentation modeling: Chi-squared automatic interaction detection. In R. P. Bagozzi, editor, *Advanced Methods of Marketing Research*, pages 118-159. Blackwell Business, Cambridge Massachusetts, 1994.
- M. Mehta, R. Agrawal, and J. Rissanen. SLIQ : A fast scalable classifier for data mining. (EDBT'96), Avignon, France, March 1996.

References (3)

- T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- S. K. Murthy, Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey, *Data Mining and Knowledge Discovery* 2(4): 345-389, 1998
- J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81-106, 1986.
- J. R. Quinlan. Bagging, boosting, and c4.5. In Proc. 13th Natl. Conf. on Artificial Intelligence (AAAI'96), 725-730, Portland, OR, Aug. 1996.
- R. Rastogi and K. Shim. Public: A decision tree classifier that integrates building and pruning. In Proc. 1998 Int. Conf. Very Large Data Bases, 404-415, New York, NY, August 1998.
- J. Shafer, R. Agrawal, and M. Mehta. SPRINT : A scalable parallel classifier for data mining. In Proc. 1996 Int. Conf. Very Large Data Bases, 544-555, Bombay, India, Sept. 1996.
- S. M. Weiss and C. A. Kulikowski. *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Morgan Kaufman, 1991.
- S. M. Weiss and N. Indurkha. *Predictive Data Mining*. Morgan Kaufmann, 1997.