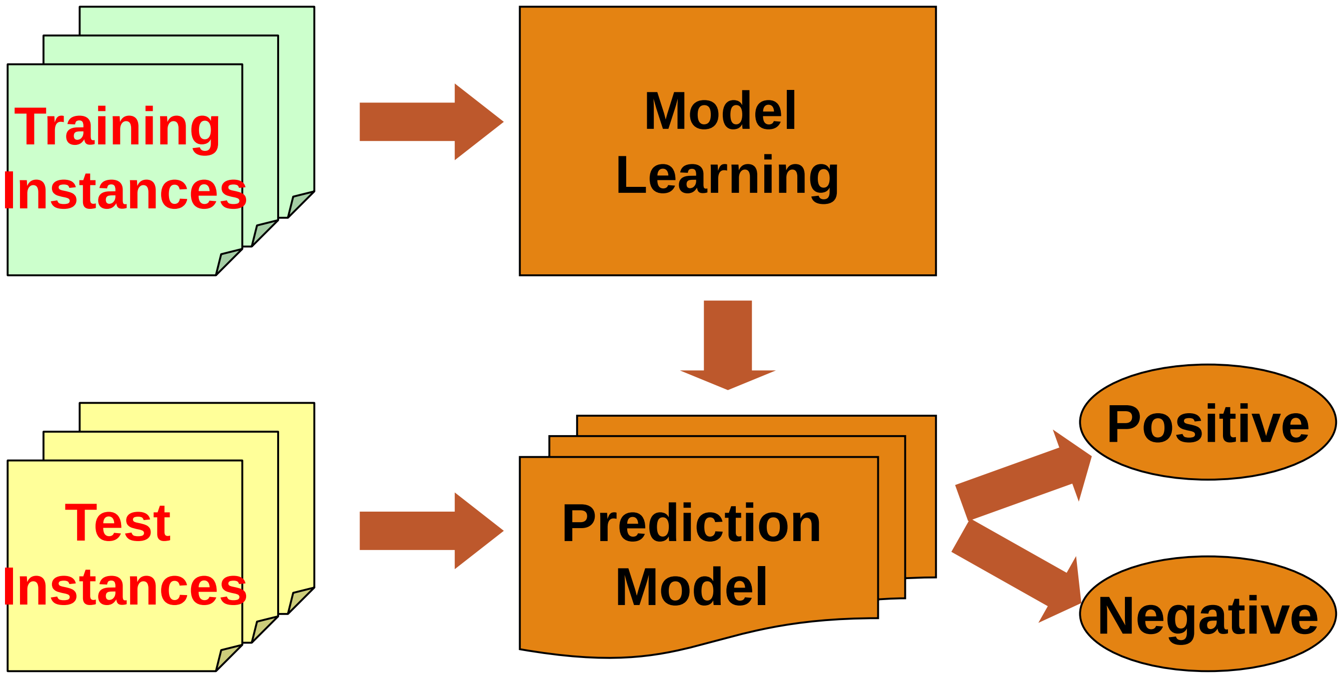
Classification in Data Mining: An Introduction

Supervised vs. Unsupervised Learning (1)

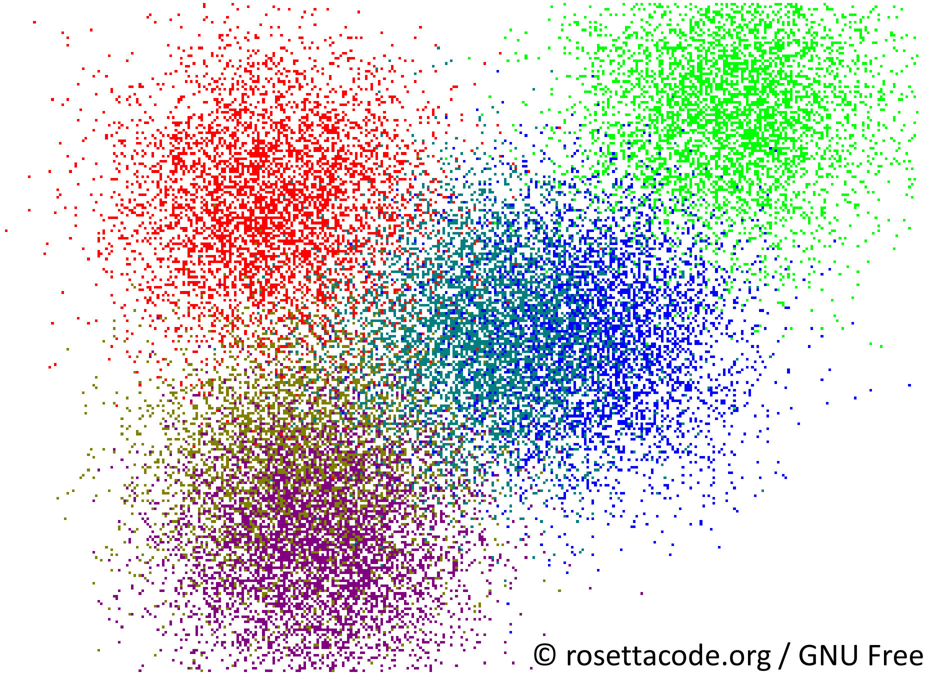
- Supervised learning (classification)
  - Supervision: The training data, such as observations or measurements, are accompanied by **labels** indicating the classes to which they belong
  - New data is classified based on the models built from the training set

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



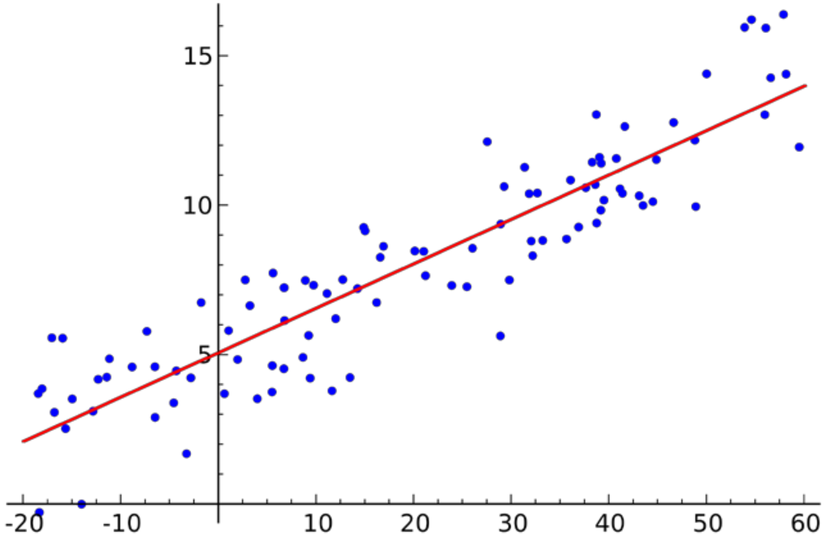
Supervised vs. Unsupervised Learning (2)

- Unsupervised learning (clustering)
  - The class labels of training data are unknown
  - Given a set of observations or measurements, establish the possible existence of classes or clusters in the data



Prediction Problems: Classification vs. Numeric Prediction

- Classification
  - Predict categorical class labels (discrete or nominal)
  - Construct a model based on the training set and the **class labels** (the values in a classifying attribute) and use it in classifying new data
- Numeric prediction
  - Model continuous-valued functions (i.e., predict unknown or missing values)
- Typical applications of classification
  - Credit/loan approval
  - Medical diagnosis: If a tumor is cancerous or benign
  - Fraud detection: If a transaction is fraudulent
  - Web page categorization: Which category it is

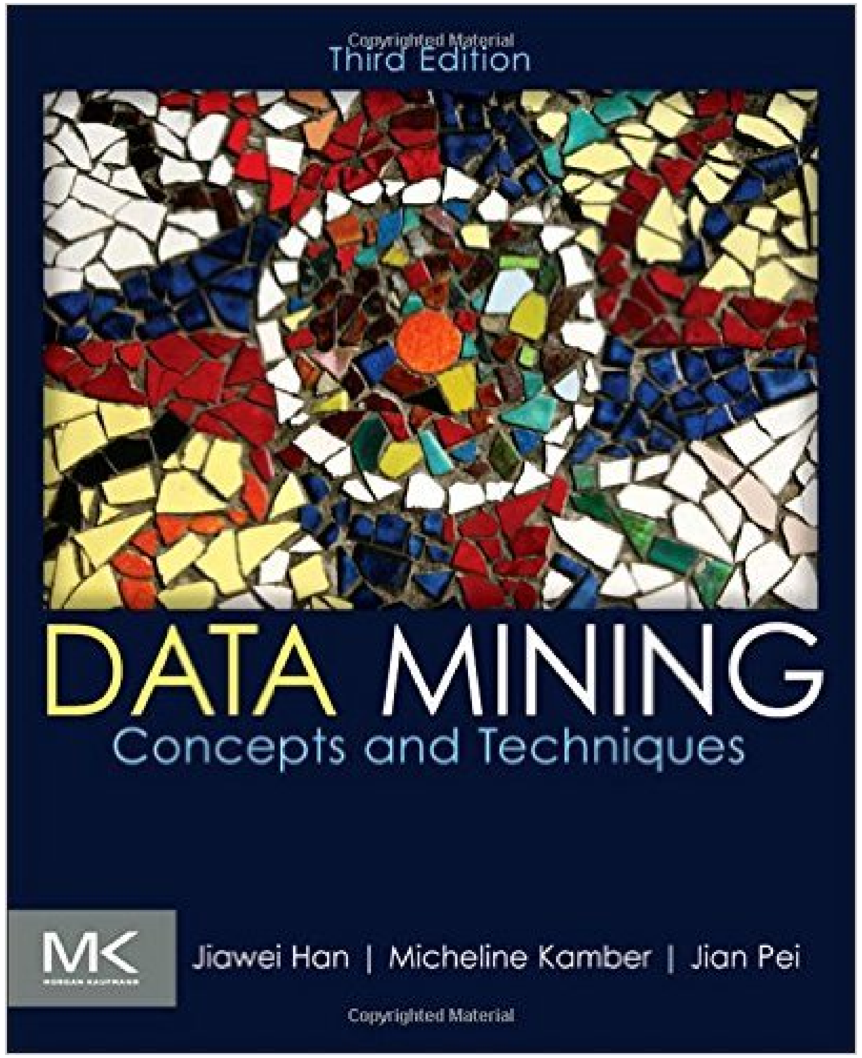


Classification—Model Construction, Validation and Testing

- **Model construction**
  - Each sample is assumed to belong to a predefined class (shown by the **class label**)
  - The set of samples used for model construction is the **training set**
  - **Model**: Represented as decision trees, rules, mathematical formulas, or other forms
- **Model validation and testing**:
  - **Test:** Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - *Accuracy:* % of test set samples that are correctly classified by the model
    - Test set is independent of training set
  - **Validation**: If *the test set* is used to select or refine models, it is called **validation** (or development) **(test) set**
- **Model deployment:** If the accuracy is acceptable, use the model to classify new data

Major Reference Readings for the Course

- Textbook
  - Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Burlington, MA: Morgan Kaufmann.
  - **Chapters most related to the course**
    - Chapter 8: Classification: Basic Concepts
    - Chapter 9: Classification: Advanced Methods
- Other references will be listed at the end of each lecture video



© 2011 Morgan Kaufmann Publishers

Course Structure

- Lesson 0: Classification in Data Mining: An Introduction
- Lesson 1: Decision Tree Induction
- Lesson 2: Bayes Classifier and Bayesian Networks
- Lesson 3: Model Evaluation, Selection, and Improvements
- Lesson 4: Linear Classifier and Support Vector Machines
- Lesson 5: Neural Networks and Deep Learning
- Lesson 6: Pattern-Based Classification and K-Nearest Neighbors Algorithm

Course General Information

- Instructor:
  - Jiawei Han, Abel Bliss Professor
  - Department of Computer Science
  - University of Illinois at Urbana-Champaign
- Teaching assistants
- Course prerequisite:
  - Familiarity with basic data structures and algorithms
- Course assessments
  - In-video questions
  - Lesson quizzes
  - Programming assignments
  - Exam

Recommended Readings

- Aggarwal, C. C. (2015). *Data mining: The textbook.* New York, NY: Springer.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). Hoboken, NJ: John Wiley.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York, NY: Springer.
- Mitchell, T. M. (1997). *Machine Learning*. Columbus, OH: McGraw Hill.
- Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2013). *Introduction to data mining* (2nd ed.). Boston, MA: Addison-Wesley.
- Weiss, S. M. & Kulikowski, C. A. (1991). *Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Burlington, MA: Morgan Kaufmann.
- Witten, I. H. & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). Burlington, MA: Morgan Kaufmann.
- Zaki, M. J. & Meira Jr., W. (2014). *Data mining and analysis: Fundamental concepts and algorithms*. Cambridge, UK: Cambridge University Press.

References

- Morgan Kaufmann. (2011). *Data mining: Concepts and techniques (3rd ed.) book cover* [Online image]. Retrieved Feb 16, 2018 from https://www.elsevier.com/books/data-mining-concepts-and-techniques/han/978-0-12-381479-1
- rosettacode.org. (2018). *Cluster diagram* [Online image]. Retrieved Feb 16, 2018 from https://goo.gl/g7KTCQ

Decision Tree Induction

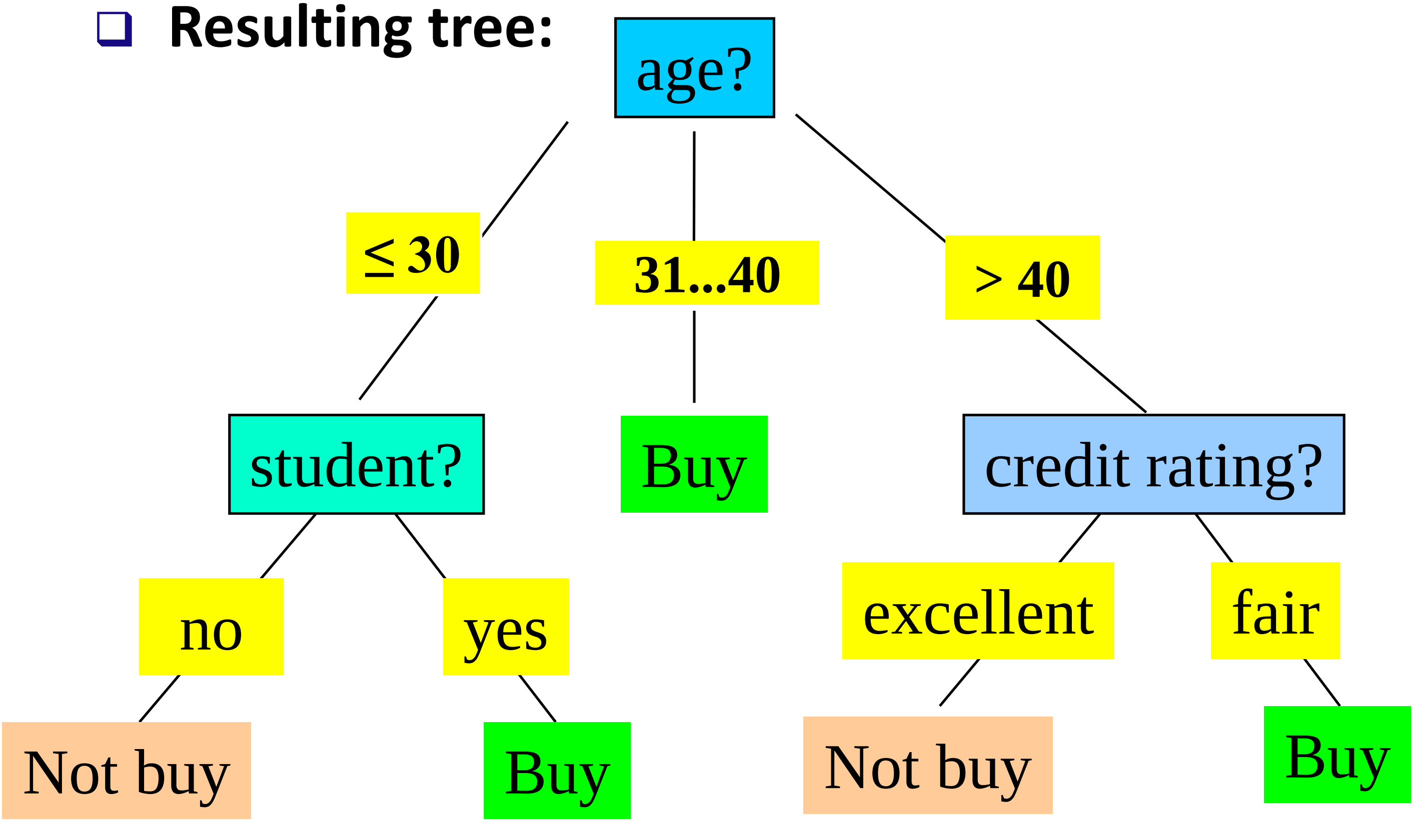
Outline

- Decision Tree Induction: Basic Idea and Algorithm
- Alternative Attribute Selection Measures in Decision Tree Induction
- Overfitting and Tree Pruning
- Decision Tree Construction in Large Datasets
- Visualization of Decision Trees and Tree Construction by Visual Data Mining

Decision Tree Induction: Basic Idea and Algorithm

Decision Tree Induction: An Example

- **Decision tree construction**:
  - A top-down, recursive, divide-and-conquer process
- **Resulting tree:**



Training data set: Who buys computer?

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

Note: The data set is adapted from "Playing Tennis" example of R. Quinlan

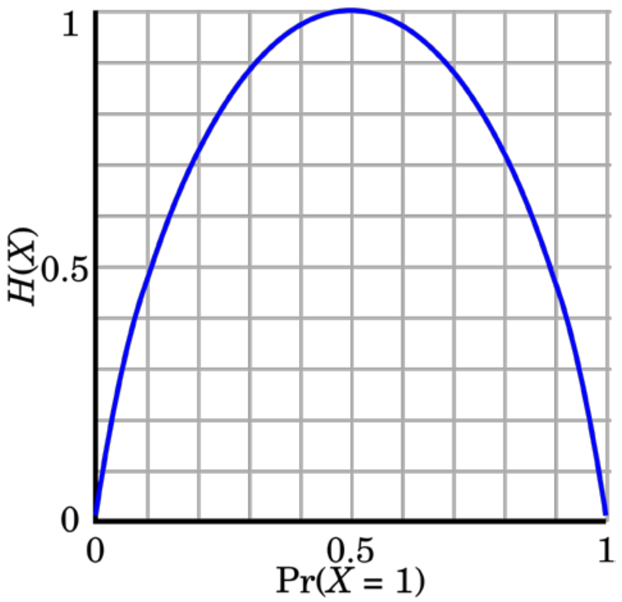
From Entropy to Info Gain: A Brief Review of Entropy

- Entropy (Information Theory)
  - A measure of uncertainty associated with a random number
  - Calculation: For a discrete random variable Y taking m distinct values $\{y_1, y_2, \ldots, y_m\}$

$$H(Y) = -\sum_{i=1}^{m} p_i \log(p_i) \quad \textit{where } p_i = P(Y = y_i)$$

  - Interpretation
    - Higher entropy → higher uncertainty
    - Lower entropy → lower uncertainty
- Conditional entropy

$$H(Y|X) = \sum_{x} p(x) H(Y|X = x)$$



m = 2

Information Gain: An Attribute Selection Measure

- Select the attribute with the highest information gain (used in typical decision tree induction algorithm: ID3)
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- Information gained by branching on attribute A:

$$Gain(A) = Info(D) - Info_A(D)$$

Example: Attribute Selection with Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…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

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

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

Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly, we can get

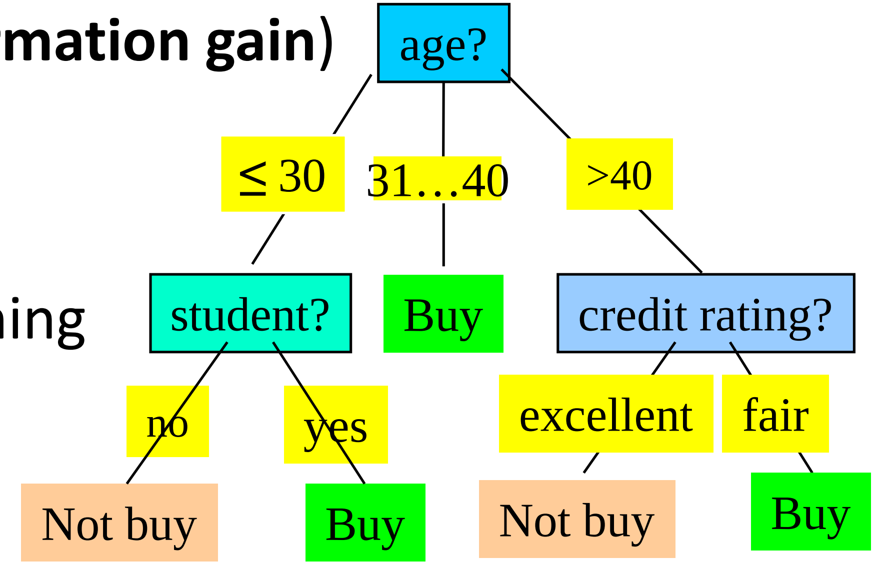
$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

Decision Tree Induction: Algorithm

- Basic algorithm
  - Tree is constructed in a **top-down, recursive, divide-and-conquer manner**
  - At start, all the training examples are at the root
  - Examples are partitioned recursively based on selected attributes
  - On each node, attributes are selected based on the training examples on that node, and 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
  - There are no samples left
- Prediction
  - **Majority voting** is employed for classifying the leaf



How to Handle Continuous-Valued Attributes?

- Method 1: Discretize continuous values and treat them as categorical values
  - E.g., age: < 20, 20…30, 30…40, 40...50, > 50
- Method 2: Determine the ***best split point*** for continuous-valued attribute A
  - Sort the value A in increasing order, E.g., 15, 18, 21, 22, 24, 25, 29, 31, …
  - *Possible split point:* The midpoint between *each pair of adjacent values*
    - $(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
    - e.g., (15+18)/2 = 16.5, 19.5, 21.5, 23, 24.5, 27, 30, …
  - The point with the *maximum information gain* for A is selected as the **split-point** for A
- Split: Based on split point P
  - The set of tuples in D satisfying $A \le P$ vs. those with $A > P$

Alternative Attribute Selection Measures in Decision Tree Induction

Gain Ratio: A Refined Measure for Attribute Selection

- Information gain measure is biased toward attributes with a large number of values
- Gain ratio: Overcomes the problem (as a 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)
  - The attribute with the maximum gain ratio is selected as the splitting attribute
  - Gain ratio is used in a popular algorithm C4.5 (a successor of ID3) by R. Quinlan
- Example
  - $\mathrm{SplitInfo}_{\mathrm{income}}(\mathrm{D}) = -\frac{4}{14}\log_2\frac{4}{14} - \frac{6}{14}\log_2\frac{6}{14} - \frac{4}{14}\log_2\frac{4}{14} = 1.557$
  - GainRatio(income) = 0.029/1.557 = 0.019

Another Measure: Gini Index

- Gini index: Used in CART, and also in IBM IntelligentMiner
- If a data set D contains examples from n classes, gini index, $gini(D)$ is defined as
  - $gini(D) = 1 - \sum_{j=1}^{n} p_j^2$
    - 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 which 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*)

Computation of Gini Index

- Example: 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
  - $gini_{income \in \{low, medium\}}(D) = \frac{10}{14} gini(D_1) + \frac{4}{14} gini(D_2)$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = 0.443$$

$$= Gini_{income \in \{high\}}(D)$$

  - $Gini_{\{low,\ high\}}$ is 0.458; $Gini_{\{medium,\ high\}}$ is 0.450
  - Thus, split on $income$ ∈ {low, medium} (i.e., also $\{high\}$) has the lowest Gini index
- The attributes discussed above assume categorical attributes
- The algorithm can also be adapted to continuous-valued attributes
  - One may need other tools, e.g., clustering, to get the possible split values

Comparing Three Attribute Selection Measures

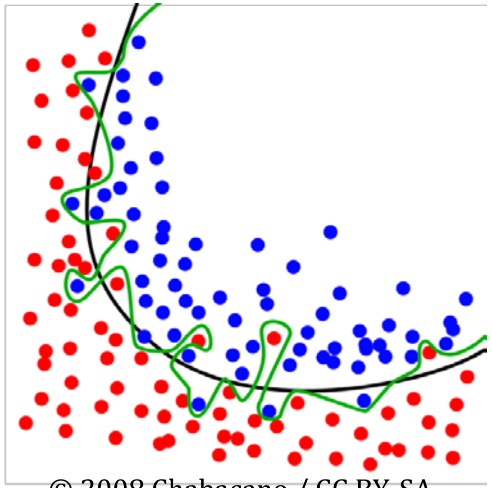
- The three measures, in general, return good results but
  - **Information gain**:
    - Is biased toward multivalued attributes
  - **Gain ratio**:
    - Tends to prefer unbalanced splits in which one partition is much smaller than the others
  - **Gini index**:
    - Is biased to multivalued attributes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

Other Attribute Selection Measures

- Minimal Description Length (MDL) principle
  - Philosophy: The simplest solution is preferred
  - The best tree is the one that requires the fewest # of bits to (1) encode the tree, and (2) encode the exceptions to the tree
- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- Multivariate splits (partition based on multiple variable combinations)
  - CART: Finds multivariate splits based on a linear combination of attributes
- There are many other measures proposed in research and applications
  - E.g., G-statistics, C-SEP
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

Overfitting and Tree Pruning

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"

Decision Tree Construction in Large Datasets

Classification in Large Databases

- Why is decision tree induction popular?
  - Relatively fast learning speed
  - Convertible to simple and easy to understand classification rules
  - Easy to be adapted to database system implementations (e.g., using SQL)
  - Comparable classification accuracy with other methods
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- **RainForest** (VLDB'98 - Gehrke, Ramakrishnan & Ganti)
  - Builds an AVC-list (attribute, value, class label)

RainForest: A Scalable Classification Framework

- The criteria that determine the quality of the tree can be computed separately
  - Builds an AVC-list: **AVC (Attribute, Value, Class_label)**
- **AVC-set** (of an attribute *X*)
  - Projection of training dataset onto the attribute *X* and class label, where counts of individual class label are aggregated
- **AVC-group** (of a node *n*)
  - Set of AVC-sets of all predictor attributes at the node *n*

age	income	student	credit_rating	comp
<=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

The Training Data

AVC-set on *Age*

Age	Buy_Computer	
	yes	no
<=30	2	3
31..40	4	0
>40	3	2

AVC-set on *Income*

income	Buy_Computer	
	yes	no
high	2	2
medium	4	2
low	3	1

AVC-set on *Student*

student	Buy_Computer	
	yes	no
yes	6	1
no	3	4

AVC-set on *Credit_Rating*

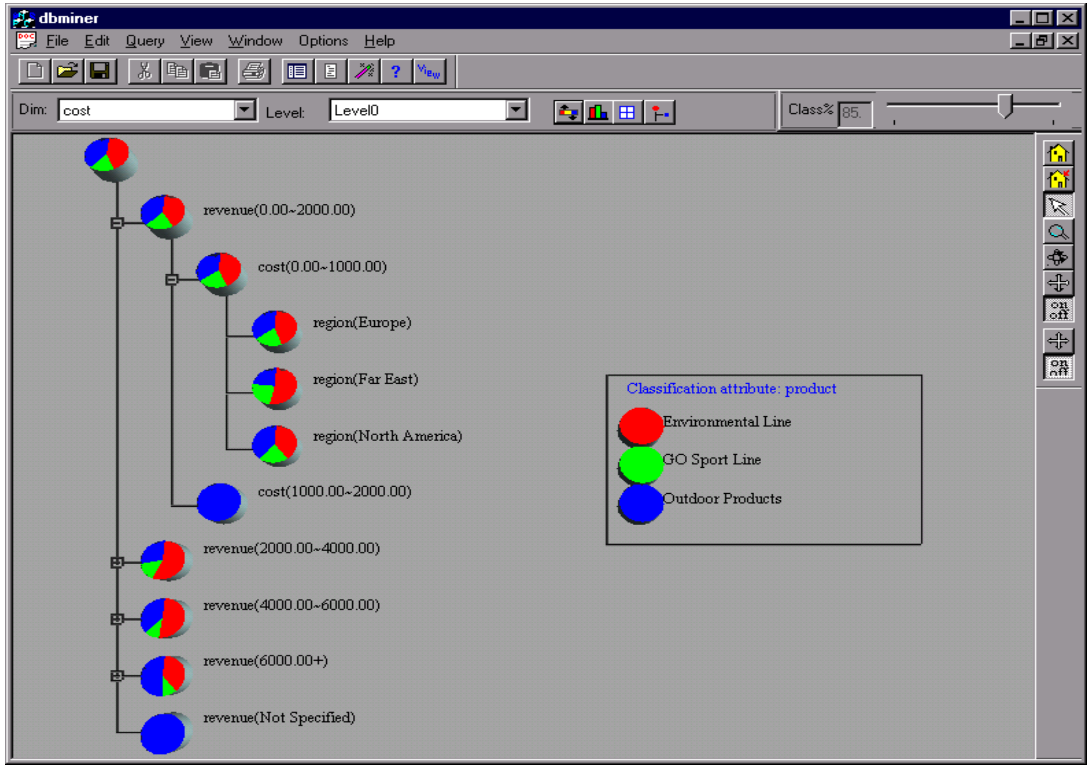
Credit rating	Buy_Computer	
	yes	no
fair	6	2
excellent	3	3

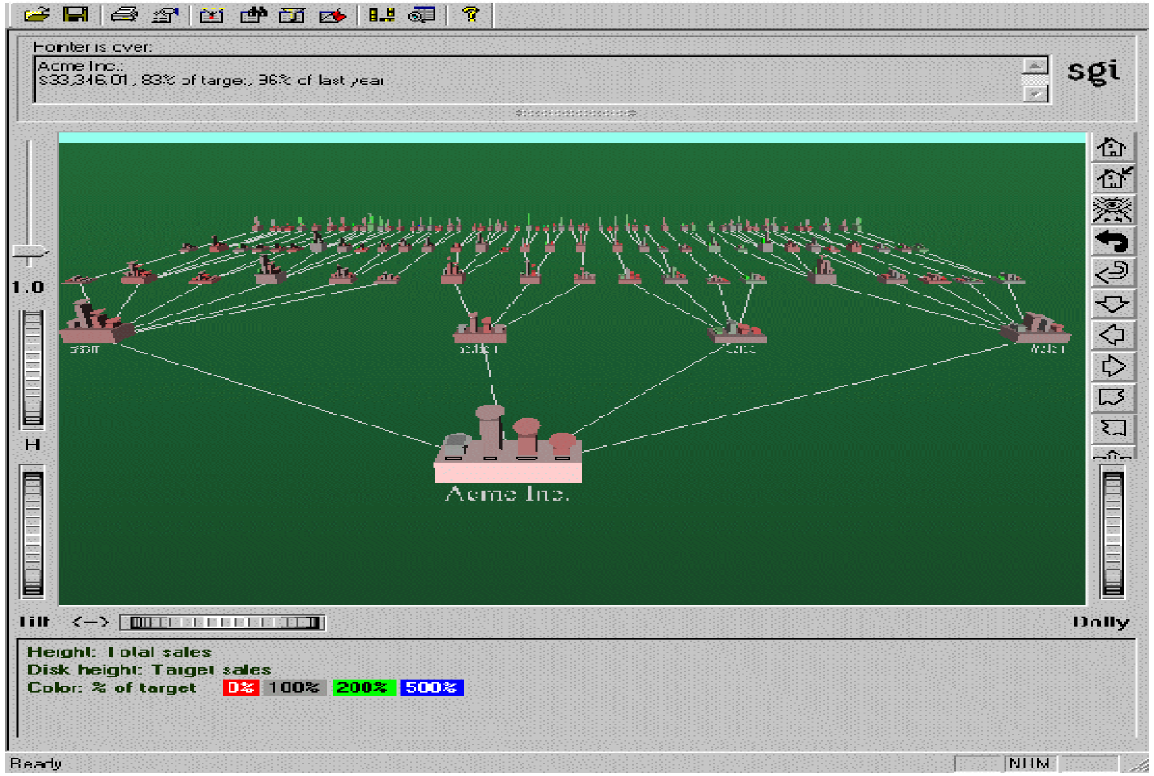
Its AVC Sets

Visualization of Decision Trees and Tree Construction by Visual Data Mining

Visualization of a Decision Tree

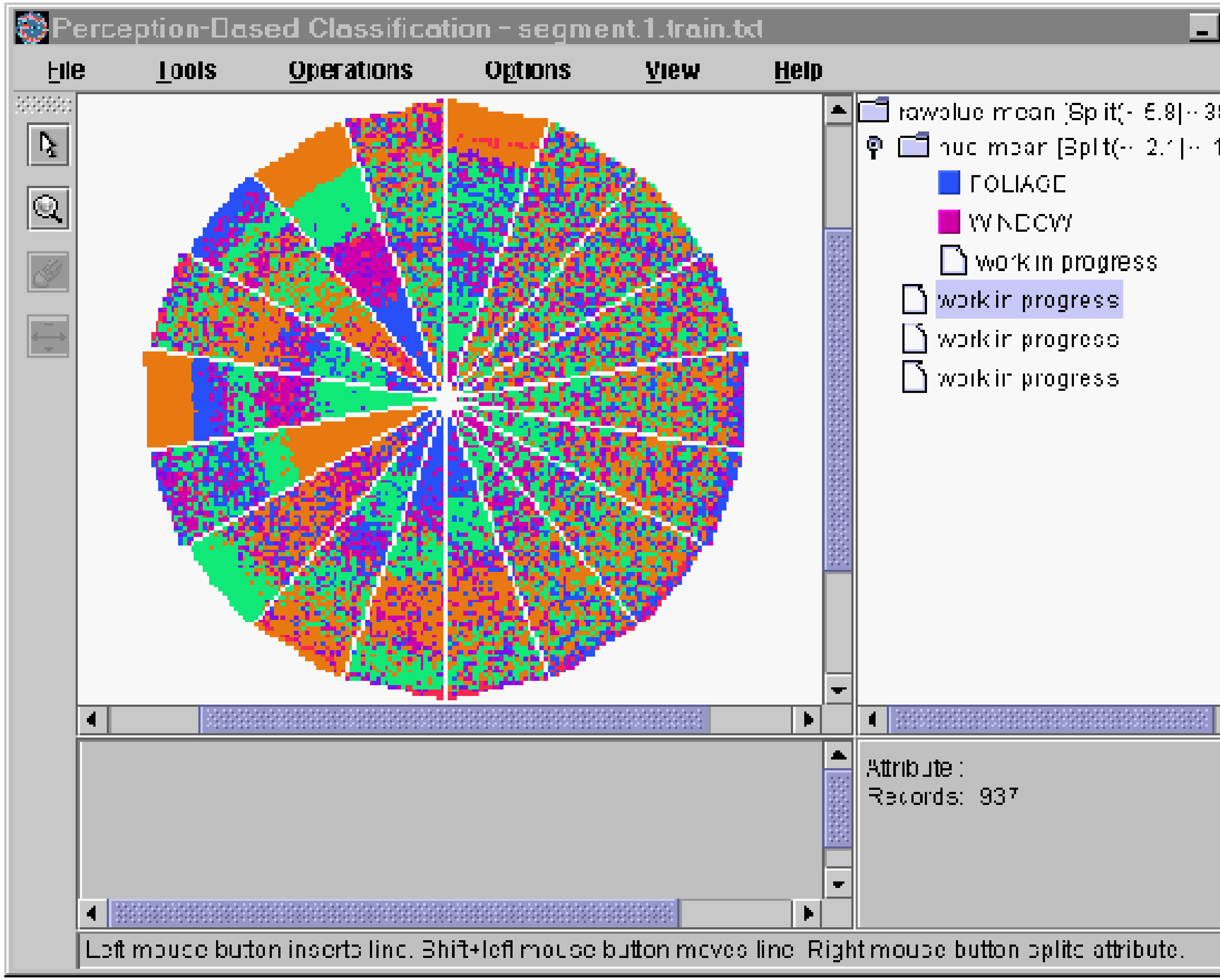
- A decision tree can be visualized with various interactive visualization tools
- The left figure is a partial tree visualization with three classes in DBMiner
- The right figure is an interactive tree visualization tool in SGI/MineSet 3.0





Interactive Visual Mining by Perception-Based Classification

- Perception-based classifier (PCB): Developed at Univ. of Munich (1999)
- One color represents one class label
- One pie represents one attribute (or variable)
- The pie with random spread implies weak classification power
- The pie with clearly partitioned color strips implies good classification power
- One can select a good attribute and regenerate new pie charts for classification at the subsequent levels



Summary

Summary

- Decision Tree Induction: Basic Idea and Algorithm
- Alternative Attribute Selection Measures in Decision Tree Induction
- Overfitting and Tree Pruning
- Decision Tree Construction in Large Datasets
- Visualization of Decision Trees and Tree Construction by Visual Data Mining

Recommended Readings

- Ankerst, M., Elsen, C., Ester, M., & Kriegel, H.P. (1999). Visual classification: An interactive approach to decision tree construction. *KDD*, 392-396. Retrieved from https://dl.acm.org/citation.cfm?id=312298
- Gehrke, J., Ramakrishnan, R., & Ganti, V. (1998). Rainforest: A framework for fast decision tree construction of large datasets. *Data Mining and Knowledge Discovery, 4*(2-3), 127-162. Retrieved from https://link.springer.com/article/10.1023/A:1009839829793
- Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning, 40*(3), 203-228. Retrieved from https://link.springer.com/article/10.1023/A:1007608224229
- Mingers, J. (1989). An empirical comparison of pruning methods for decision tree induction. *Machine Learning, 4*(2), 227-243. Retrieved from https://link.springer.com/article/10.1023/A:1022604100933
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81-106. Retrieved from https://link.springer.com/content/pdf/10.1007/BF00116251.pdf
- Quinlan, J. R. (1993). C4.5: Programs for machine learning. *Machine Learning, 16*(3), 235-240. Retrieved from https://link.springer.com/article/10.1007/BF00993309

References

- Brona, Alessio Damato, & Rubber Duck. (2007). *Binary entropy plot.svg* [Online image]. Retrieved Feb 16, 2018 from https://goo.gl/DtnBNp
- Chabacano. (2008). *Overfitting.svg* [Online image]. Retrieved Feb 16, 2018 from https://goo.gl/3R7xZG