Data Mining

Andrew Kusiak
Intelligent Systems Laboratory
2139 Seamans Center
The University of Iowa
Iowa City, Iowa 52242 - 1527

Tel: 319 - 335 5934 Fax: 319 - 335 5669
andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak

Partially based on the material provided
by J Han and M Kamber

Classification and Prediction

Outline

- Learning
- Classification and prediction
- Classification by decision tree induction
- Classification by backpropagation
- Other Classification Methods
- Prediction
- Classification accuracy

http://www.kdnuggets.com/

Learning

What is learning?

- Extraction of knowledge
- Pattern creation

- Basis of learning
  - Training data set

Classification vs. Prediction

Categorical

- Classification:
  - predicts categorical class labels
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it for classifying cases with unknown outcomes

Continuous

- Prediction:
  - models continuous-valued functions, i.e., predicts unknown or missing values

http://www.twocrows.com/glossary.htm

Classification and Decision Making

Step 1: Model construction (e.g., knowledge extraction)

Learning

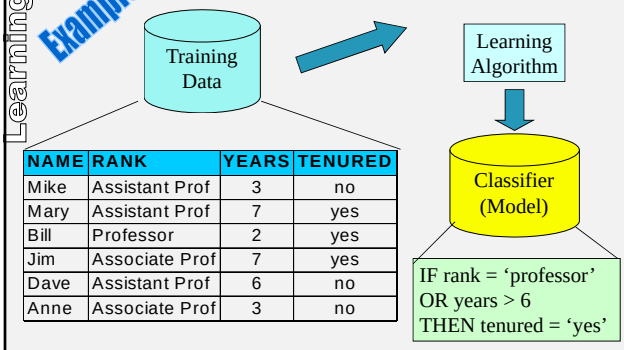
Step 2: Model usage (e.g., decision making)

Expert system

Classification Process: Model Construction

Learning

Example 1

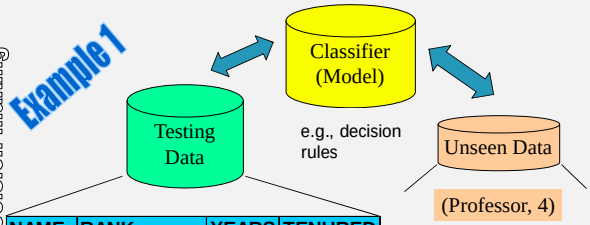


NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classification Process: Use the Model in Prediction

Decision making

Example 1



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	6	yes
Joseph	Assistant Prof	7	yes

Training Data Set

Example 2

No.	F1	F2	F3	F4	D
1	1.02	Red	2.98	High	Good
2	2.03	Black	1.04	Low	Bad
3	0.99	Blue	3.04	High	Good
4	2.03	Blue	3.11	High	Good
5	0.03	Orange	0.96	Low	Bad
6	0.04	Blue	1.04	Medium	Bad
7	0.99	Orange	1.04	Medium	Good
8	1.02	Red	0.94	Low	Bad

Extracted Rules

Example 2

No.	F1	F2	F3	F4	D
1	1.02	Red	2.98	High	Good
2	2.03	Black	1.04	Low	Bad
3	0.99	Blue	3.04	High	Good
4	2.03	Blue	3.11	High	Good
5	0.03	Orange	0.96	Low	Bad
6	0.04	Blue	1.04	Medium	Bad
7	0.99	Orange	1.04	Medium	Good
8	1.02	Red	0.94	Low	Bad

Learning

Rule 1. IF (F4 = High) THEN (D = Good); [1, 3, 4]

Rule 2. IF (F4 = Medium) AND (F2 = Blue)
THEN (D = Bad); [6]

Rule 3. IF (F4 = Medium) AND (F2 = Orange)
THEN (D = Good); [7]

Rule 4. IF (F4 = Low) THEN (D = Bad); [2, 5, 8]

Patterns

Example 2

Rule 1. IF (F4 = High) THEN (D = Good); [1, 3, 4]
Rule 2. IF (F4 = Medium) AND (F2 = Blue)
THEN (D = Bad); [6]
Rule 3. IF (F4 = Medium) AND (F2 = Orange)
THEN (D = Good); [7]
Rule 4. IF (F4 = Low) THEN (D = Bad); [2, 5, 8]

No.	F1	F2	F3	F4	D	Rule
1	1.02	Red	2.98	High	Good	1
2	2.03	Black	1.04	Low	Bad	4
3	0.99	Blue	3.04	High	Good	1
4	2.03	Blue	3.11	High	Good	1
5	0.03	Orange	0.96	Low	Bad	4
6	0.04	Blue	1.04	Medium	Bad	2
7	0.99	Orange	1.04	Medium	Good	3
8	1.02	Red	0.94	Low	Bad	4

Decision Making

Example 2

No.	F1	F2	F3	F4	D
9		Blue		Medium	?

New case

Extracted knowledge

No.	F1	F2	F3	F4	D	Rule
1	1.02	Red	2.98	High	Good	1
2	2.03	Black	1.04	Low	Bad	4
3	0.99	Blue	3.04	High	Good	1
4	2.03	Blue	3.11	High	Good	1
5	0.03	Orange	0.96	Low	Bad	4
6	0.04	Blue	1.04	Medium	Bad	2
7	0.99	Orange	1.04	Medium	Good	3
8	1.02	Red	0.94	Low	Bad	4

DM Algorithm

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

Classification by Decision Tree Induction

- Decision tree
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution
- Decision tree generation consists of two phases
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

Decision Tree Algorithm

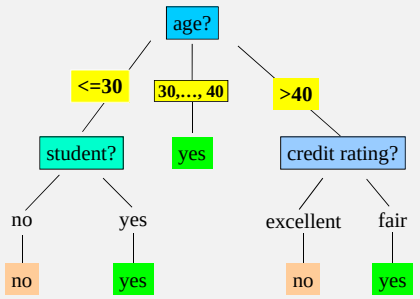
Quinlan's data set

Training data set

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

http://www.megaputer.com/products/pa/algorithms/dt.php3

Output: A Decision Tree for "*buys_computer*"



Buys computer

Information Gain (C4.5)

Attributes (Features)

Set S of examples (rows)

p elements (rows, examples) — P class

n elements (rows) — N class

Information

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information Gain in Decision Tree Induction

Homogeneity measure

- Assume that using attribute A a set *S* will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$, i.e., *v* attribute A values
  - If S_i contains p_i examples of *P* and n_i examples of *N*, the entropy, or the expected information needed to classify objects in all sub-trees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

- The encoding information that would be gained by branching on *A*

$$Gain(A) = I(p,n) - E(A)$$

Example

Training data set

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

Continuous attribute values → a split value

Attribute Selection by Information Gain Computation

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

Entropy $E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$

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

$Gain(age) = I - E = 0.246$

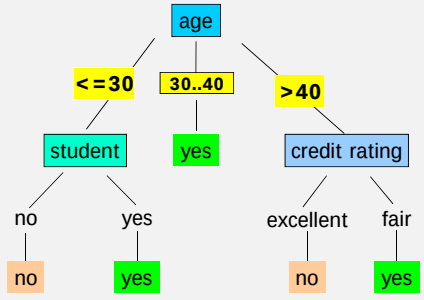
Also

$Gain(income) = 0.029$

$Gain(student) = 0.151$

$Gain(credit\_rating) = 0.048$

Decision Tree Revisited



Examples @ each terminal node

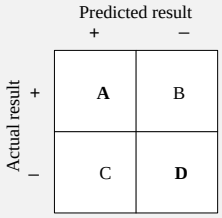
Classification Accuracy 1

DM definition

Classification accuracy (CA) of a rule set is the ratio of the number of correctly classified objects from the test set and all objects in the test set

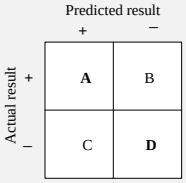
Classification Accuracy 2

Biostatistics



Accuracy = (A + D)/(A + B + C + D)

Classification Accuracy 3



Accuracy = (A + D)/(A + B + C + D)

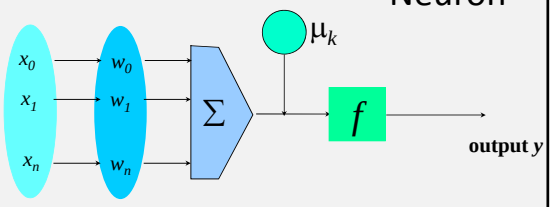
- Sensitivity (true positive rate) = A/(A+B)
- Specificity (true negative rate) = D/(C+D)
- False negative rate = B/(A+B) = 1 – Sensitivity (Type I error)
- False positive rate = C/(C+D) = 1 – Specificity (Type II error)
- Positive predicted value = A/(A+C)
- Negative predicted value = D/(B+D)

Neural Networks

- Advantages
  - prediction accuracy is generally high
  - robust, works when training examples contain errors
  - output may be discrete, real-valued, or a vector of several discrete or real-valued attributes
  - fast evaluation of the learned target function
- Disadvantages
  - long training time
  - difficult to understand the learned function (weights)
  - not easy to incorporate domain knowledge

http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

Neuron



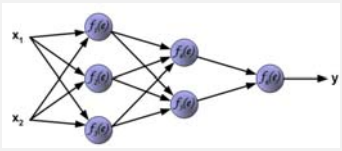
Input vector x **weight vector w** **weighted sum** **Activation function**

- The n-dimensional input vector x is mapped into variable y by means of the scalar product and a nonlinear function mapping
- Perceptron learning demo: http://www.lri.fr/~marc/EEAAX/Neurones/tutorial/perceptron/html/index.html

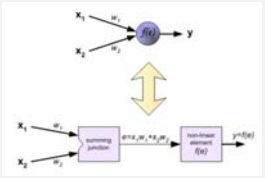
Neural Network Training

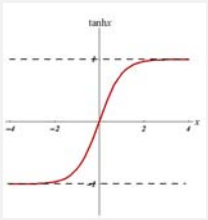
- The ultimate objective of training
  - obtain a set of weights that makes almost all the tuples in the training data classified correctly
- Steps
  - Initialize weights with random values
  - Feed the input tuples into the network one by one
  - For each unit
    - Compute the net input to the unit as a linear combination of all the inputs to the unit
    - Compute the output value using the activation function
    - Compute the error
    - Update the weights and the bias

NN Training: the Neuron

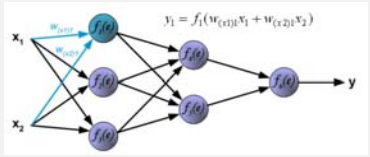


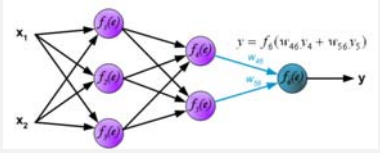
F(e) is a sigmoidal function, typically tanh, or logistic function





NN Training: the Network





NN Training: Back Propagation

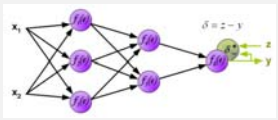


Figure 1

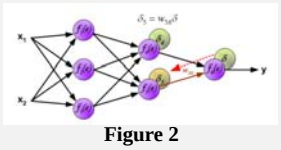


Figure 2

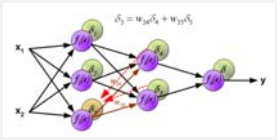


Figure 3

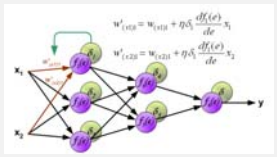


Figure 4

Other Classification Methods

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

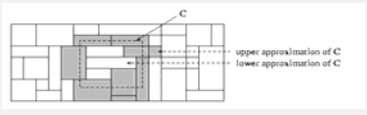
Genetic Algorithms

- GA: based on an analogy to biological evolution
- Each rule is represented by a string of bits
- An initial population is created consisting of randomly generated rules
  - e.g., IF A_1 and Not A_2 then C_2 can be encoded as 100
- Based on the notion of survival of the fittest, a new population is formed to consists of the fittest rules and their offspring
- The fitness of a rule is represented by its classification accuracy on a set of training examples
- Offspring are generated by crossover and mutation

http://www4.ncsu.edu/eos/users/d/dhloughl/public/stable.htm

Rough Set Approach

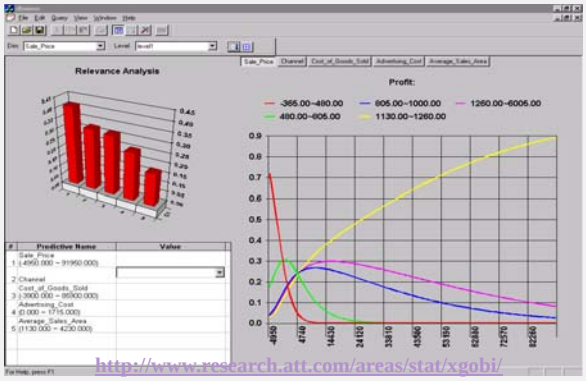
- Rough sets are used to approximately or "roughly" define equivalent classes
- A rough set for a given class C is approximated by two sets: a lower approximation (certain to be in C) and an upper approximation (cannot be described as not belonging to C)
- Finding the minimal subsets (reducts) of attributes (for feature reduction) is NP-hard but a discernibility matrix is used to reduce the computation intensity



Regression Analysis and Log-Linear Models in Prediction

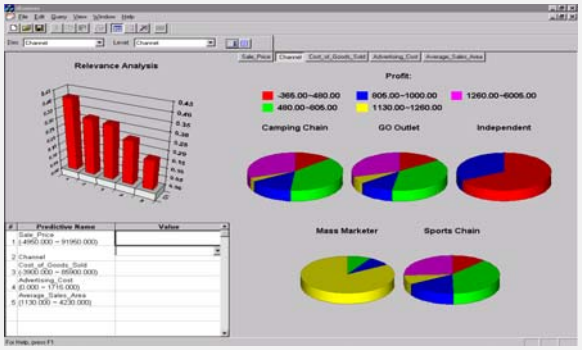
- Linear regression: $Y = \alpha + \beta X$
  - Two parameters , α and β specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of Y_1, Y_2, …, X_1, X_2, ….
- Multiple regression: $Y = b0 + b1\ X1 + b2\ X2$.
  - Many nonlinear functions 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: $p(a, b, c, d) = \alpha_{ab} \beta_{ac} \chi_{ad} \delta_{bcd}$

Prediction: Numerical Data



http://www.research.att.com/areas/stat/xgobi/

Prediction: Categorical Data



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 sub-sample as test data --- *k*-fold cross-validation
  - for data set with moderate size
- Bootstrapping (leave-one-out)
  - for small size data

Reference

Han J. and M. Kamber (2000), *Data Mining: Concepts and Techniques*, Academic Press, San Diego, CA.