Data Mining: Overview of DM methods

Ch 5, 6 and 7 from Hand's book

Course outline

- Introduction (1)
- Data Warehouse (1)
- Data Preprocessing (2/1)
- Classification Methods
- Clustering Methods (4)
- Pattern finding (2)
- Applications (1-2)
- Multimedia Mining (2)
- Survey of recent research (2)
  - Presentations
- Course project (2)
  - Presentations

Paper presentations

- I will post a list of paper with tentative assignment by tomorrow (04/10) morning.
  - Each paper will be assigned to a student for a 20 min presention
  - Two other "reviewers" will also be assigned.
- We can discuss and re-assign if necessary on Monday. Please contact me in advance if you have any specific issues.

Classification Algorithms

- READING: C10 (Hand), C7 (Han)
- We will be covering the following
  - Linear discriminants and Perceptrons
  - Decision tree induction
  - Bayesian Classification
  - Nearest neighbor methods
- Today: Before getting into the details→a quick look at the components of DM/classification methods (C5,C6 and C7 of Hand)

Overview of DM methods

- Data mining components
- Models and patterns
- Curse of dimensionality
- Scoring functions

DM Algorithms: components

- Task: visualization, classification, clustering, regression,….
- Structure: functional form of the model we are fitting to the data. E.g., linear regression, hierarchical clustering, etc.
- Score function: to judge the quality of the fitted models. E.g., misclassification error, squared error, etc.
- Search or optimization methods: computational methods used to find the score functions; e.g., greedy search.
- Data management techniques: for storing, indexing and retrieving data.

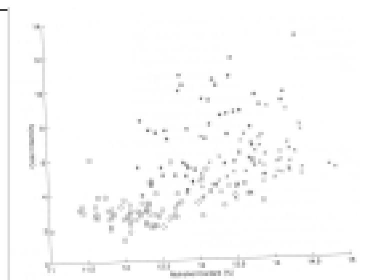
Examples

	CART	BP	A Priori
Task	Classification/ regression	Regression	Rule pattern discovery
Structure	Decision tree	NN	Association rules
Score function	Cross-validated loss function	Squared error	Support/ accuracy
Search	Greedy search	Gradient descent	Breadth-first with pruning
DM Method	--	--	Linear scans

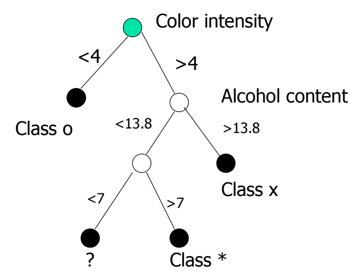
CART

- CART: Classification and Regression Trees
- CART is widely used for producing classification and regression models with a tree based structure.
  - Task: prediction (classification)
  - Model structure: tree
  - Score function: cross-validated loss function
  - Search method: greedy local search
  - Data management: unspecified

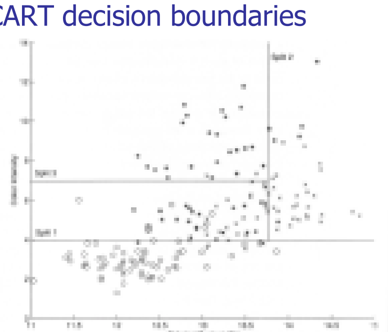
Scatter plot: color vs alcohol content



CART based classification



CART decision boundaries



CART

- At the root node, CART picks the best variable for splitting the data into two groups.
- This splitting procedure is then recursively applied to the data in each of the child nodes.
- There is a final "pruning" process.
- Score function: misclassification loss function

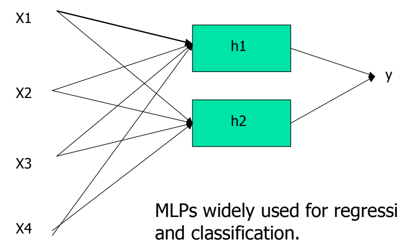
$$\sum_{i=1}^{n} C(y(i), \hat{y}(i))$$

Is the loss incurred when the class label for the i-th data vector, y(i) is predicted by the tree to be y(\hat). C is specified usually by an m x m matrix, m is the # classes.

CART Software

- http://www.salford-systems.com
- Developed at Stanford and UC-Berkeley
- CART is a registered trademark
- Other decision trees: ID2, C4.5 etc
  - See http://www.cse.unsw.edu.au/~quinlan
  - Or his company http://www.rulequest.com

Multi-layer perceptrons

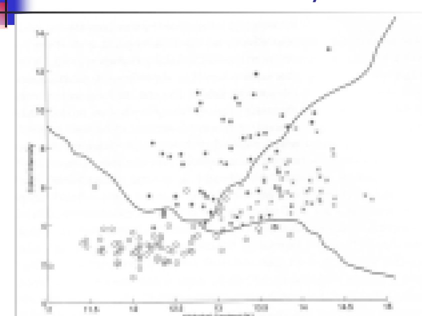


MLPs widely used for regression and classification.

MLP

- Task = prediction: classification or regression
- Structure = multiple layers of non-linear transformations of weighted sums of the inputs
- Score function = sum of squared errors
- Search method = steepest descent from randomly chosen initial parameter values
- Data management technique = online or batch

MLP decision boundary



MLP

- "weights" are updated using learning methods such as *Back-propagation*.
- No widely accepted procedure for determining the structure of the MLP (mostly ad-hoc rules)
  - CART structure is automatically learnt
- Training can be computationally expensive

A Priori algorithm for Association Rule Learning

- An association rule is a simple probabilistic statement about the co-occurrence of certain events in a database
- Eg: IF A=1 AND B=1 THEN C=1 with probability *p*.
- The conditional probability *p* is referred to as the *accuracy* or *confidence* of the rule
- Prob (A=1, B=1, C=1) is the support

Association Rules

- Originated in *market-basket* data analysis
  - Task=description: association between variables
  - Structure=probabilistic "association" rules
  - Score function=thresholds on accuracy and support
  - Search method=systematic search (exponential number of possibilities!)
  - DM technique=multiple linear scans

Vector Space methods for Text Retrieval

- General task of retrieval by content: given a query object and a large database of objects, we would like to find the k objects in the database that are most similar to the query object
- Eg., query=short list of keywords, database="web pages";
- An important issue: how is similarity defined?

Text retrieval

- Reduce documents to a uniform vector represention
- Let t1, t2, .., t*p* be *p* terms. A document (a row in our data matrix) is represented by a vector of length *p*, where the *i*-th component contains the count of how often the term t*i* appears in the document.
- Similarity: angle between the two vectors in the p space.

Text retrieval

- Task=retrieve k most similar documents
- Representation=vector of term occurrences
- Score function=angle between two vectors
- Search method=various techniques
- DM technique=various fast indexing strategies

Overview of DM methods

- Data mining components
- Models and patterns
- Curse of dimensionality
- Scoring functions

Predictive models

- Model building in data mining is data-driven
- It seeks to capture the relationships in the data
- In a predictive model, one of the variables is expressed as a function of the others.

$$\hat{y} = f(x_1, ..., x_p; \theta)$$

θ represents the parameters of the model

- If Y is quantitative, then the mapping of p dimensional X to Y is known as regression
- if Y is categorical, → classification

Regression models with linear structure

$$\hat{Y} = a_0 + \sum_{j=1}^{p} a_j X_j$$

$$\theta = \{a_0, ..., a_p\}$$

- Geometrically, this model describes a p-dim hyperplane embedded in a (p+1)-dim space with the slope determined by the a_js and intercept by a_0.
- Goal of parameter estimation is to choose the "a" values to locate and angle this hyperplane so as to provide the best fit to the data {x(i), y(i)}.

Generalizations

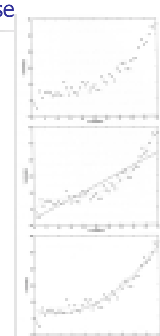
$$\hat{Y} = a_0 + \sum_{j=1}^{p} a_j f_j(X_j)$$

- f_j are functions, possibly smooth and possibly nolinear (log, square root, etc.)
- Further generalizations to allow cross-product terms
- Note that these above models are *nonlinear* in the variables X but are still *linear* in the parameters→ parameter estimation is still simple.

$Y=0.001x^3 - 0.05x^2 + x$ + noise

(a) 50 data points simulated according to third order polynomial
(b) A linear fit to the data
(c) Fit to the model $Y=ax^2+bx+c$. dotted lines are the true model from which the data is generated.

Model parameters estimated by minimizing the sum of squared errors between the predicted and the true values.



issues

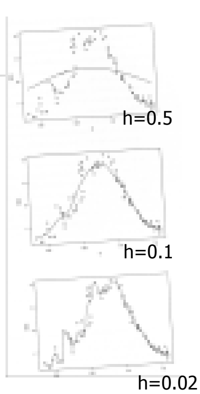
- As the "dimensionality" *p* increases, estimation becomes difficult.
- Instead of transforming the predictor variables, one can transform the response variables (Y) instead.
  - Of course, we do not know in advance what such transformations should be. (that is why data mining is *interesting* and *challenging).*
- Generalized linear models (neural networks)—more later.
- Local piecewise model structures

Nonparametric local models

Kernel estimators: $K\left(\frac{x-z}{h}\right)$

K is a smoothing function that determines the contribution to the estimate at a new point *z* from a data set point at *x*.

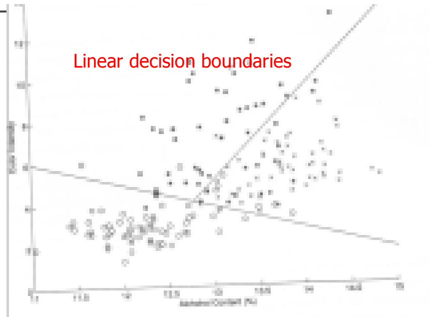
The size of this contribution will depend on both K and the bandwidth h.



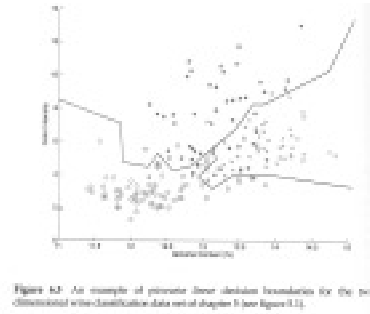
Kernel methods

- Closely related to *nearest neighbor methods*
- In NN methods, the data determines the bandwidth by defining it in terms of the *number* of nearest neighbors
- Note that all these methods are nonparametric as the model is largely data driven (except for the choice of *h*—the bandwidth parameter)
  - Lack of *interpretability* of the model

Predictive models for classification



Piecewise linear



Overview of DM methods

- Data mining components
- Models and patterns
- Curse of dimensionality
- Scoring functions

Curse of dimensionality

- Most classification / clustering / regression / indexing methods do not generalize well to higher dimensions.
- Higer dimensions – p = 10 to p=1000….
- Eg.: estimating parameters for a normal distribution such that the error is less than 0.1 at x=0; data simulated with zero mean and unit covariance matrix. The number of data points needed to achieve this accuracy grows exponentially with the dimension p.
  - P=1, 4 points; p=2, 19 points; p=3, 67 points; p=6→2790; p=10→ 842K.

Curse of dimensionality

Two obvious strategies

- Use a subset of relevant variables to construct a model;
  - Some X variables completely unrelated to Y (eg. Date of birth of a person vs credit worthiness)
  - Others may be redundant (total sales and sales tax)
- Transform the original p variables into a new set of p'<<p variables
  - E.g. using PCA, NN, etc.

Variable selection

- Difficult to identify which variables are dependent on which, given a finite sample size
- Various measures can be used
  - Correlation of X w.r.t Y
  - If Y is categorical, average mutual information between Y and X'

$$I(Y;X') = \sum_{i,j} p(y_i, x_j^{'}) \log \frac{p(y_i, x_j^{'})}{p(y_i)\,p(x_j^{'})}$$

In general, subset selection methods rely on heuristic search to find good model structures

Transformations for HD data

- Replace the observed variables with a smaller set of variables
- Projection Pursuit Regression

$$\hat{y} = \sum_{j=1}^{p'} w_j h_j \left(\alpha_j^T \mathbf{x}\right)$$

Procedures for determining the w, h and the projection directions (\alpha_j) can be complex.

- Principal Component Analysis

Overview of DM methods

- Data mining components
- Models and patterns
- Curse of dimensionality
- Score functions

Score functions for DM Algorithms

- Purpose: to rank models as a function of how useful the models are to the data miner
  - Score functions for models vs patterns
  - SF for Predictive structures vs descriptive structures
  - SF for Models of fixed complexity vs models of different complexity

Scoring patterns

- There is no general consensus on how patterns should be scored
  - One person's noisy outlier might be another's jackpot
- Patterns might be evaluated in terms of how "interesting" or "unexpected" they are to the data analyst—but this requires prior knowledge
- Consider IF a THEN b with probability p; How interesting or informative this rule is to an uninformed observer?

Scoring patterns

- IF a THEN b with probability p

Assume that P(b) is known (this is the marginal probability of event b). For eg., if P(b)=0.25 and p=P(b|a)=0.75, then it is interesting

So a simple measure could be |P(b|a) – P(b)|

Judging the novelty and utility of a pattern is often quite subjective and application specific

Score functions: Predictive Models

Let $D = \{(x(1), y(1)), ..., (x(n), y(n))\}$; Let $\hat{f}(x(i), \theta)$ be the prediction generated by the model, using parameter values θ.

Sum of squared errors: $$S_{SSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{f}(x(i);\theta) - y(i)\right)^2$$

Misclassification rate $$S_{0/1}(\theta) = \frac{1}{N}\sum_{i=1}^{N} I\left(\hat{f}(x(i);\theta), y(i)\right)$$

$I(a,b) = 1$ if $a \neq b$; $=0$ otherwise

Score functions: Descriptive models

- Descriptive models: no target variables to be predicted➔less clear how define a score fn.
- Examples include: models for the overall probability distribution of the data (density estimation), partitioning into groups (clustering), modeling the relationship between variables (dependency modeling)

$$\hat{p}(\mathbf{x};\theta): \text{ prob of observing a data point } \mathbf{x}$$

X is assumed categorical.
Better models assign higher probability to observed data.

maximize $L(\theta)=\prod_{i=1}^{n}\hat{p}(\mathbf{x}(i);\theta)$

Scoring descriptive models

$L(\theta)=\prod_{i=1}^{n}\hat{p}(\mathbf{x}(i);\theta)$ ➔likelihood function; convenient to work with log

$$\log L(\theta)=\sum_{i=1}^{n}\log\hat{p}(\mathbf{x}(i);\theta),\text{ or equivalently}$$

$$\text{minimize } S_L(\theta)=-\log L(\theta)=-\sum_{i=1}^{n}\log\hat{p}(\mathbf{x}(i);\theta)$$

Notes:
-log P is the error term—gets larger as P gets smaller.
Max of P is 1 ➔ lower bound on S_L=0.
This score function is quite general.
Limitations: Outliers dominate the cost…good or bad?

Summary

- Data mining components
- Models and patterns
- Curse of dimensionality
- Scoring functions

Next: Classification Methods