Data Mining

Henryk.Maciejewski@pwr.edu.pl
WS 2015/16

1

Data Mining – Contents of this Part

- Purpose and Definition of Data Mining
- Application Areas of Data Mining
- Data Mining Process → Lab class
- Data Mining Algorithms in Detail
  (predictive modelling, clustering, association rules,…)

2

Data Mining – Further Reading

- J. Han, M. Kamber, Data Mining: *Concepts and Techniques, Third Edition*
- T. Hastie, R. Tibshirani, J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*

What is Data Mining – Introductory Example

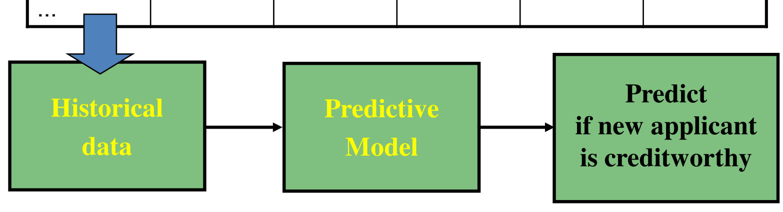
- Business problem: loan risk prediction

  ? Is new applicant *creditworthy*?

- Solutions:
  - Judgement by trained/experienced evaluator
  - Data Mining based solution: generate rules as to who is likely to be creditworthy based on bank's *historical data*

Example cntd.

Income	Job	Age	Marital status	...	Target
2450	Self	34	M		Bad
1600	Other	55	M		Good
4800	Mgr	39	S		Good
...					



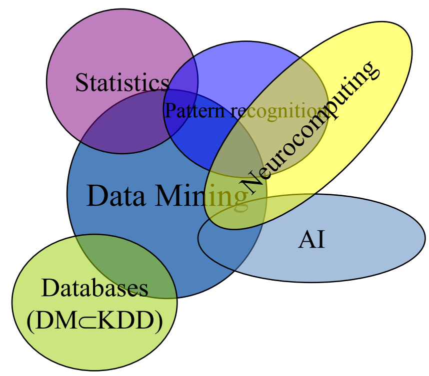
Definitions of DM

- *„Advanced methods for exploring and modelling relationships in large amounts of data"*
- *„Discovery of useful summaries of data"*
- *„Process of identifying useful patterns and regularities in large bodies of data"*

- Originally, DM was what statisticians taught us *not to do*...
  (DM = drawing invalid inferences from data by using invalid methods, or drawing conclusions true for purely statistical reasons)

DM as Multidisciplinary Area

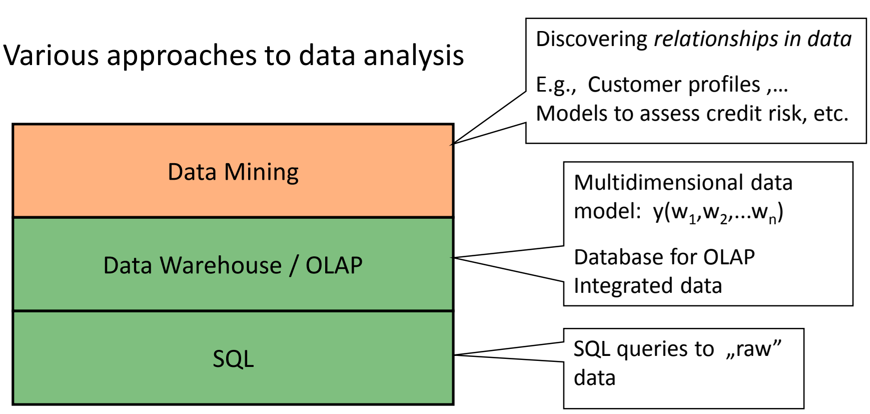
- Statistics
- AI (machine learning)
- Research in clustering algorithms
- Visualization techniques
- Pattern recognition
- Neurocomputing
- Databases (DM $\subset$ KDD)



7

Data Mining vs. Data Warehousing

- Various approaches to data analysis

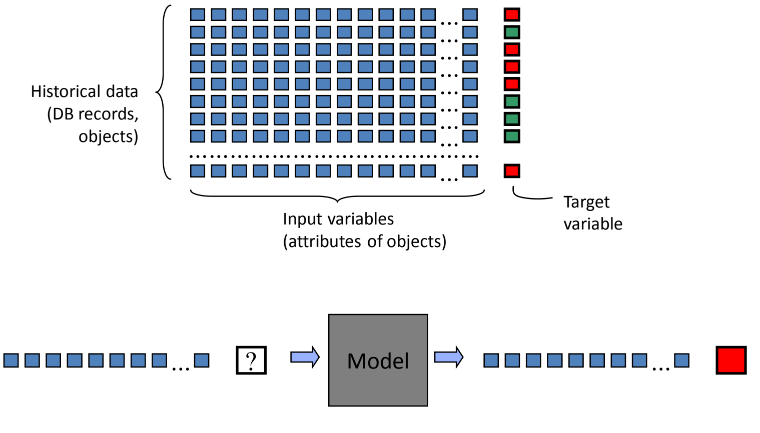


8

Data Mining Techniques

- Predictive modelling
- Cluster analysis
- Dependency derivation („*association rules*”)
- Web mining
- Text mining
- Sequence matching
- Time series forecasting
- …

Predictive Modelling – Problem Formulation and Applications



Predictive Modelling – Problem Formulation

- Predictive modelling involves
  1. Building *a model* of relationship
     target(input variables)
  2. Estimation of predictive performance of this model for *new data*

- Predictive modelling:
  - Classification – for qualititive targets
  - Regression – for quantitative targets

11

Predictive Modelling – Problem Formulation

- Methods/algorithms used for predictive modelling
  - Linear regression
  - Linear, nonlinear discriminant analysis
  - Logistic regression
  - Classification and regression trees
  - Perceptron algorithm, neural networks
  - Support vector machines
  - Nonparametric classifiers (nearest neighbours)
  - …

12

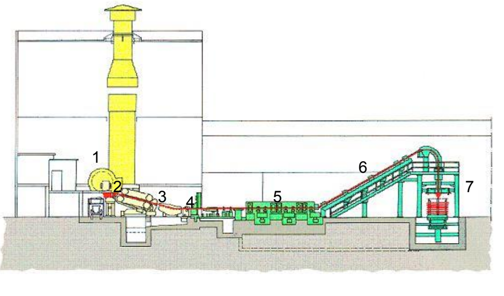
Predictive Modelling – Applications

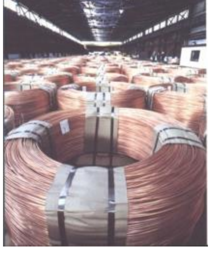
- Finance:
  - Credit risk analysis
  - Credit card analysis
- Insurance – fraud detection
- Marketing – campaign planning, targeted marketing
- Churn analysis (telecoms)
- Genetics/medicine
  - risk group prediction, breast cancer recurrence risk assessment, prediction of response to chemiotherapy, tumour classification, etc.
- Classification of text documents (sentiment analysis, spam detection, topic/subject classification,…)
- Manufacturing
  - product quality assessment, process monitoring
- …

13

Example: Product Quality Assessment

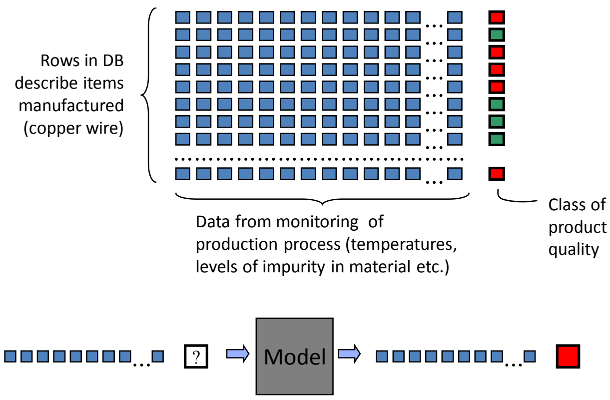
- Based on data from industrial process monitoring, assess quality of products
- Example: production of copper wire

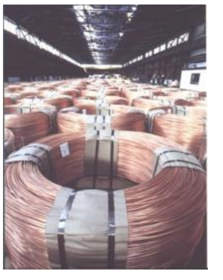




14

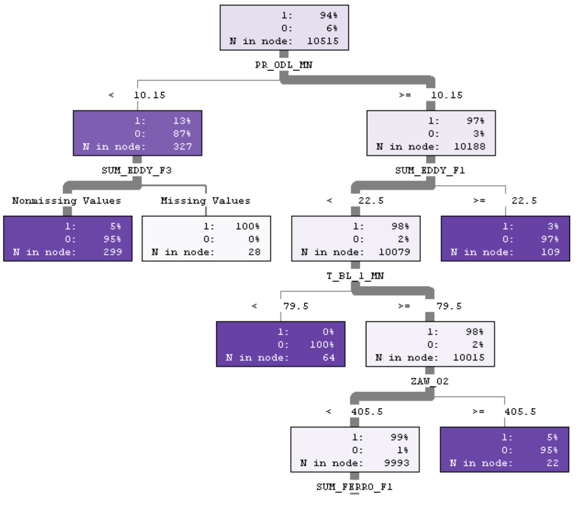
Example: Product Quality Assessment





15

Example: Classification Tree for QA

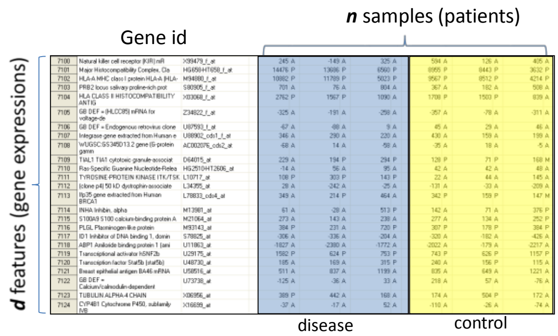


16

Bioinformatics: Analysis of *Massive Throughput* Experiments

- Gene expression DNA microarrays – expression of thousands of genes of a patient/tissue measured in single experiment

- Challenge: ***d>>n***
  (in „typical" DM studies, *d<<n*)

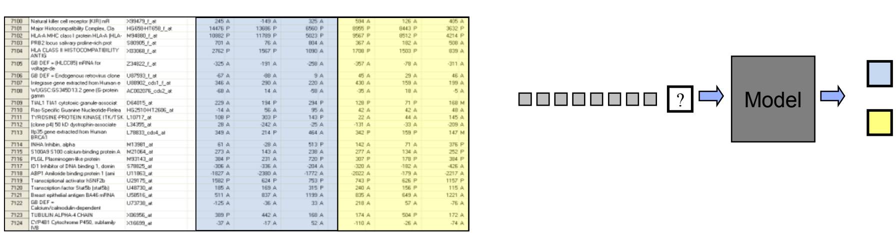


17

Class Prediction Based on Microarray Data

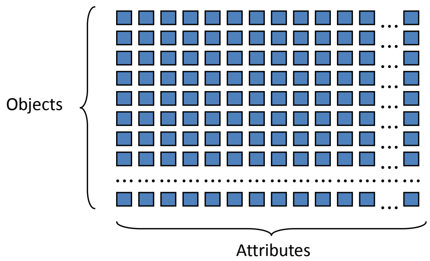
- Medical diagnostics, selection of best therapies, disease risk analysis, agriculture,…
- Examples of medical tests based on microarray assays:

  MammaPrint – breast cancer recurrence risk assessment
  Oncotype DX – test if cancer is likely to respond to chemotherapy



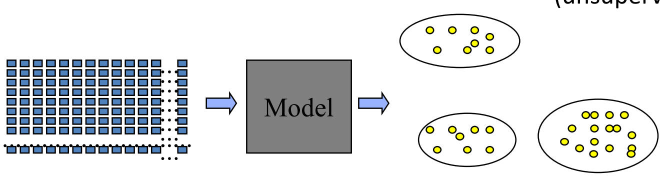
18

Clustering – Formulation



Find groups of objects *similar* to each other and *dissimilar* from other groups

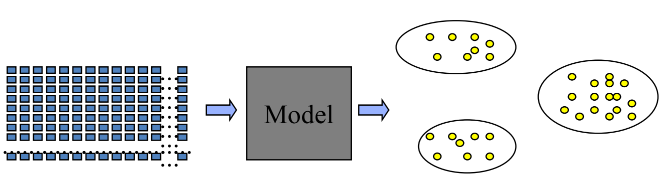
No target variable (unsupervised learning)



19

Clustering – Applications

- Customer profiling
- Target marketing
- Bioinformatics
  - discovery of similarly expressed genes
  - disease taxonomy (groups of patients with similar expression patterns)

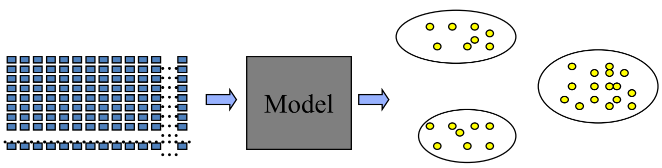


20

Clustering – Methods

- Hierarchical clustering
- k-means (nondeterministic)
- Vector Quantization
- SOM
- …

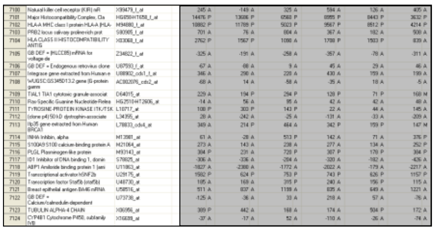
- Challenges
  - No robust methods to determine the *right* number of clusters
  - Results depend on method of clustering

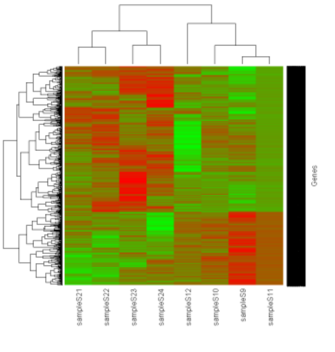


21

Example: Clustering Gene Expressions

- Class discovery
  - Clustering by rows – discovery of relationships between genes
  - Clustering by columns – disease taxonomy

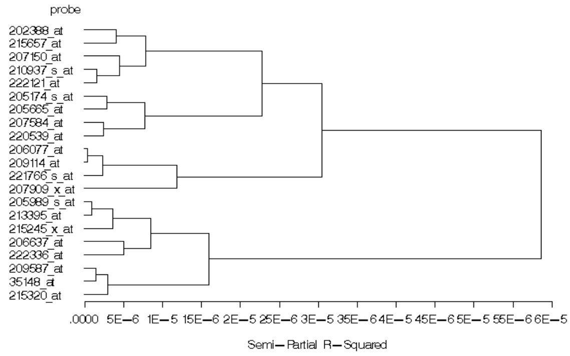




22

Example: Clustering Gene Expressions

- Dendrogram – results of hierarchical clustering (hierarchy of clusters)



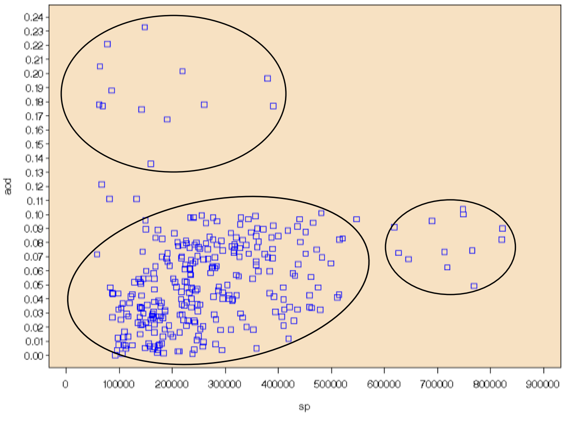
Example: SkyCat Project

- Digital Palomar Observatory Sky Survey (POSS II)
- 3000 images, 23040x23040x16 bit/pixel = 3TB
- Ca 5x10^{7} galaxies, 2x10^{9} stars, 10^{5} quasars
- Objects too blurred – classification by human astronomer impossible
- For some objects high resolution CCD images available

- Task: automatic classification of objects (assign to one of 4 classes)

- Solution based on:
  - Decision trees (ID3, GID3*)
  - *Unsupervised classification* algorithms
- Results: ~95% of correctly classified objects

Example: Profiling Restaurants



- Analysis of chain of fast food restaurants
- Each restaurant represented by value of sales (sp) vs. production losses (aod)
- Three clusters = profiles identified

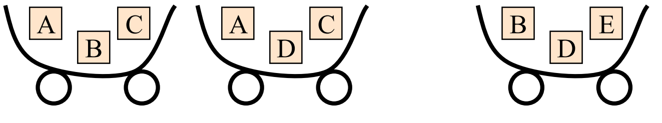
25

Association Rules

- Task: find groups of items frequently bought together (also known as *frequent itemset mining*)
- Results are usually expressed as *association rules*:

  A→B

  where {A,B} is *frequent enough*
  Pr{B|A} is *high enough*

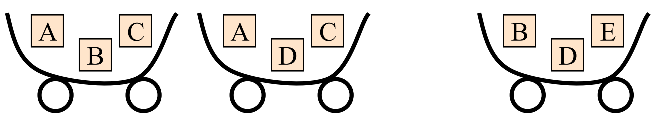


26

Association Rules

- **Support** of rule = frequency of {A,B}
- **Confidence** of rule = Pr(B|A)

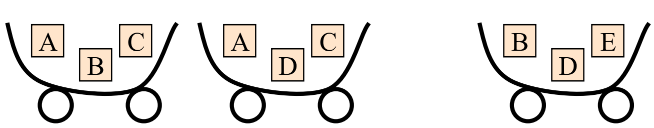
- Another interesting parameter of a rule is **lift** = Pr(B|A) / Pr(B)
  Pr(B)=prob. of buying B, no rule known
  Interesting rules significantly *lift* probability of buying B



27

Association Rules

- The task is to find *all rules* with given min support and min confidence
- Task is challenging due to large volume of data to be searched, e.g.,
  - 10^5 items in store
  - 20 millions transactions per day



28

Association Rules – Applications

- Baskets = documents
- Articles = words

  Words appearing together may imply phrases characteristic to some area. Used for automatic text classification.

- Baskets = sentences
- Articles = documents

  Documents with many identical sentences may suggest plagiarism or mirror sites in the Internet

29

Association Rules - Applications

- Recommendation systems, e.g., amazon.com

30

Time Series Forecasting

- Input data: time series variable(s) (spaced equally over time)
- Data assumed to have
  - Trend
  - Seasonal behaviour
  - Noise
- Forecasting done by extrapolating trends in past values of the series

level	year
1120	1871
1160	1872
963	1873
1210	1874
1160	1875
1160	1876
813	1877
1230	1878
1370	1879
1140	1880
995	1881
935	1882
1110	1883
994	1884
1020	1885
960	1886
1180	1887
799	1888
958	1889
1140	1890
1100	1891
1210	1892
1150	1893
1250	1894
1260	1895
1220	1896
1030	1897
1100	1898

31

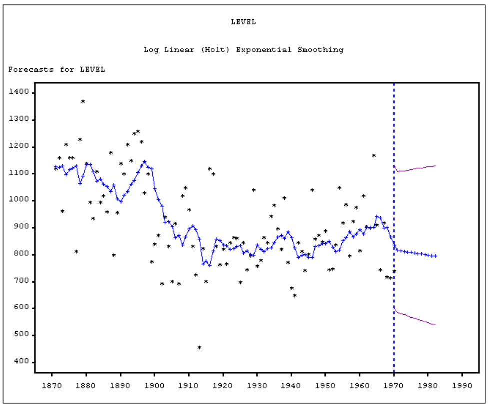
Time Series Forecasting

- Applications:
  - Short term electricity consumption demands
  - Economics - sales data, share prices, employment figures
  - Meteorology – rainfall, temperature patterns
  - …
- Several methods to model the trends:
  - ARIMA,
  - Exponential smoothing models,
  - AutoRegressive Trees

level	year
1120	1871
1160	1872
963	1873
1210	1874
1160	1875
1160	1876
813	1877
1230	1878
1370	1879
1140	1880
995	1881
935	1882
1110	1883
994	1884
1020	1885
960	1886
1180	1887
799	1888
958	1889
1140	1890
1100	1891
1210	1892
1150	1893
1250	1894
1260	1895
1220	1896
1030	1897
1100	1898

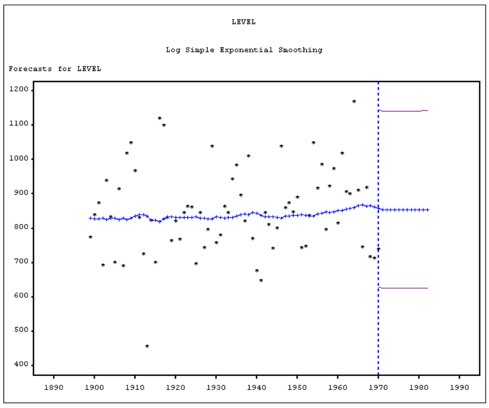
32

Example – Flow of the Nile River



- Prediction based on annual volume of the Nile measured at Aswan
- Shift in level in 1899 due to construction of the new dam at Aswan (and partly weather changes)

Example – Flow of the Nile River



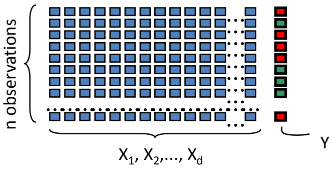
- This forecast is based on more recent data only (data before 1899 removed)

Algorithms for Predictive Modelling – Contents

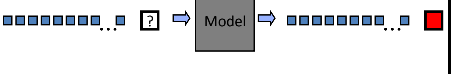
- Regression
- Classification
- Auxiliary topics:
  - Estimation of prediction error
  - Reduction of dimensionality

Predictive Modelling – Terminology

- Given input data
  $\{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_n,y_n)\}$
  1. find model of relationship between Y and $X_1,X_2,\ldots,X_d$
  2. estimate predictive performance of the model for *new data*



- X_i – input variable (other names: independent variable, predictor, regressor, explanatory variable, carrier, factor, covariate)
- Y – target variable (also response)



Linear Methods for Regression

- We seek Y(X) – assuming that the relationship is *linear* (assumption simplifies computations to fit the model)
- Many nonlinear problem can be modelled with linear regression – by applying transformations to variables
- We will discuss the following problems
  - Fitting the model to data
  - Verifying goodness-of-fit
  - Should the model include *all* the features X_1 through X_d, or only the *best features*? (especially important for high-dimensionality data)

37

Linear Methods for Classification

- Regression can also used for classification – logistic regression
- Linearity assumption also important in classification (e.g. Linear Discriminant Analysis (LDA), separating hyperplanes (perceptron algorithm))

38

Theoretical Background – Statistical Decision Theory

- Notation
  - **X** $\in$ R^d – input variables (random variables)
  - Y $\in$ R – output variable (random variable)
  - Pr(**X**,Y) – joint probability distribution
- We look for a function f(X) for predicting the value of Y

39

Theoretical Background – Statistical Decision Theory

- Criterion: the function should minimize the *squared error*:

$$\mathrm{EPE}(f) = \mathrm{E}(Y - f(X))^2 = \int \left(y - f(x)\right)^2 \Pr\left(dx, dy\right)$$

- Solution – *regression function*:

$$f(x) = \mathrm{E}(Y \mid X = x)$$

40

Theoretical Background – Statistical Decision Theory

- Notice: if criterion is to minimize the L_1 norm

$$\mathrm{E}\left|Y - f(X)\right|$$

then the solution is

$$f(x) = \mathrm{median}(Y \mid X = x)$$

41

From Theory to Practice…

- Regression function f(x)=E(Y|X=x) is based on known joint probability distribution of X and Y
- How to estimate *f* from *data*: $\{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_n,y_n)\}$?
- Different approaches:
  - Parametric – build model of $f(x)$
  - Nonparametric

42

Linear Regression

- We assume that f(x)=E(Y|X=x) is a linear function of $X_1, X_2, \ldots, X_d$:

$$f(X) = \beta_0 + \sum_{j=1}^{d} X_j \beta_j$$

β - vector of unknown model coefficients

- As X_j we can take:
  - Quantitative input variables
  - Nonlinear transformations of inputs (e.g., log)
  - Polynomial terms, e.g., $X_2=X_1^2$, etc.
  - Numerically coded levels of *qulitative* variable

43

Fitting the Model

- Estimation of $\beta=[\beta_0, \beta_{1,\ldots,} \beta_d]$ based on $\{(\boldsymbol{x}_1,y_1),\ldots, (\boldsymbol{x}_n,y_n)\}$ – by minimization of residual sum of squares RSS(β):

$$\mathrm{RSS}(\beta) = \sum_{j=1}^{n} \left(y_j - f(x_j)\right)^2$$

- Solution:

$$\beta = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T y \qquad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \ldots \\ \boldsymbol{x}_n \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ \ldots \\ y_n \end{bmatrix}$$

44

Verifying the Model

- Goodness of fit must be verified before we attempt use the model for prediction
  How to check if the model fits the data well?

- Regression procedures in software packages (SAS, SPSS, etc.) offer several tools to do this:
  - diagnostic plots,
  - hypothesis tests if parameters significant,
  - measures of residual error, etc. – we illustrate these by example

45

Linear Regression – Example

- How weight of children depends on height and age?
- Solution – using SAS procedure reg:

```
proc reg data=kids;
  model  weight=height age;
  plot weight*age weight*height;
  plot r.*p.;
run;
```

Fit model
w = f(h,a)

Diagnostic plots for model verification

age	height	weight
14.3	56.3	85
15.5	62.3	105
15.3	63.3	108
16.1	59	92
19.1	62.5	112.5
17.1	62.5	112
18.5	59	104
14.2	56.5	69
16	62	94.5
14	53.8	68.5
13.9	61.5	104
17.8	61.5	103.5
15.7	64.5	123.5
14.9	58.3	93
14.3	51.3	50.5
14.5	58.8	89
19.1	65.3	107
15	59.5	78.5
14.7	61.3	115

46

Linear Regression – Example

- Solution:

  weight = -127.8 + 3.09 x height + 2.4 x age

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-127.8199	12.09900	-10.56	<.0001
height	1	3.09005	0.25734	12.01	<.0001
age	1	2.40275	0.55103	4.36	<.0001

Linear Regression – Example

- Overall measures concerning the fit:
  - Root MSE – mean square error in regression
  - R-Square – regression accounts for 63% of variance in data is explained by the regression model (→1 implies that the model is appropriate)

Root MSE	11.86836	**R-Square**	0.6305
Dependent Mean	101.30802	**Adj R-Sq**	0.6273
Coeff Var	11.71512		

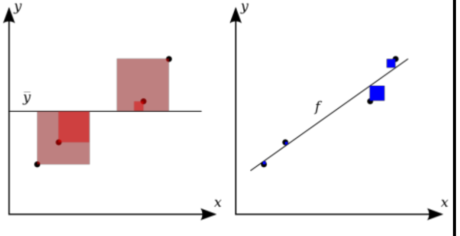
Linear Regression – Example

$$SS_{total} = \sum_{j=1}^{n} \left(y_j - \bar{y}\right)^2$$

$$SS_{reg} = \sum_{j=1}^{n} \left(f(x_j) - \bar{y}\right)^2$$

$$SS_{err} = RSS = \sum_{j=1}^{n} \left(y_j - f(x_j)\right)^2$$

$$R^2 = 1 - \frac{SS_{err}}{SS_{total}}$$



SS_{total} SS_{err}

49

Linear Regression – Example

- Testing if parameters of the model are significant
  - t Value – test of hypothesis H0: $\beta_j = 0$
  - p value < 0.05 – reject H0 → model parameters are significant

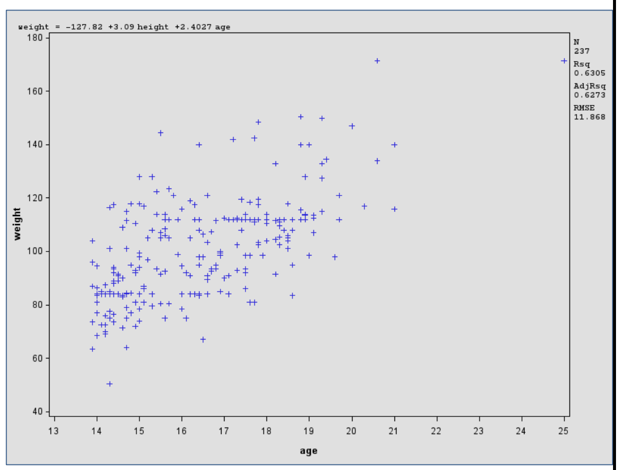
Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-127.8199	12.09900	-10.56	<.0001
height	1	3.09005	0.25734	12.01	<.0001
age	1	2.40275	0.55103	4.36	<.0001

50

Linear Regression – Example

- Diagnostic plots – visually verify linearity assumption

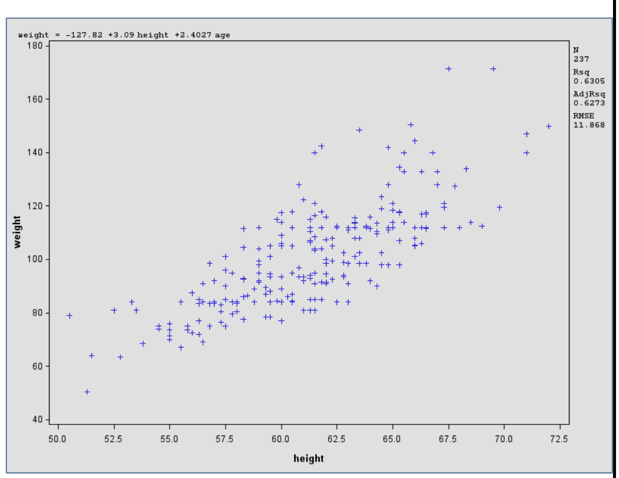
  weight vs age



Linear Regression – Example

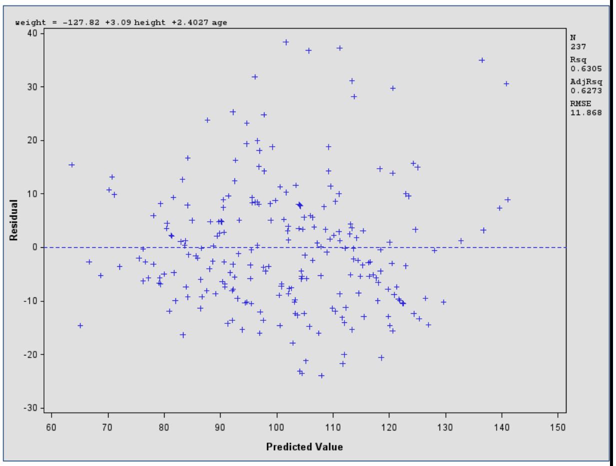
- Diagnostic plots – visualy verify linearity assumption

  weight vs height



Linear Regression – Example

- Diagnostic plots – Residual vs predicted values
- Residual = $y_{observed} - y_{predicted}$
- Trend in shape may indicate model inadequacy



Linear Regression – Finding Best Regressors

- Problem: regression based on *all features* X vs regression based only on *the best features*?

```
proc reg data=kids;
  model  weight=height age / selection=forward;
  model  weight=height age / selection=backward;
  model  weight=height age / selection=stepwise;
run;
```

- Forward – subsequent parameters added to model
- Backward – starting with complete model, parameters are removed
- Stepwise – similar to forward (but parameter can be removed after being added to the model)

Linear Regression – Finding Best Regressors

Forward Selection: Step 1
Variable **height** Entered: R-Square = 0.6004 and C(p) = 20.0137

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	53555	53555	353.14	<.0001
Error	235	35639	151.65526		
Corrected Total	236	89194			

Test H0 that „smaller" model is appropriate

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-132.99101	12.49370	17184	113.31	<.0001
height	3.81815	0.20318	53555	353.14	<.0001

55

Linear Regression – Finding Best Regressors

Forward Selection: Step 2
Variable **age** Entered: R-Square = 0.6305 and C(p) = 3.0000

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	56233	28117	199.61	<.0001
Error	234	32961	140.85795		
Corrected Total	236	89194			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-127.81991	12.09900	15721	111.61	<.0001
height	3.09005	0.25734	20309	144.18	<.0001
age	2.40275	0.55103	2678.22612	19.01	<.0001

56

Regression *vs* nonparametric approach

<u>Linear regression</u>

$f(x)$=E(Y | X=x) – assumption: linear function of $X_1,\ldots,X_d$

$$f(X) = \beta_0 + \sum_{j=1}^{d} X_j \beta_j$$

<u>Nearest neighbours method</u>

$$\hat{f}(x) = \text{Average}(y_i \mid x_i \in N_k(x))$$

- $N_k(x)$ – neighbourhood of x containing k points closest to x
- Attempt to estimate $f(x)$=E(Y | X=x) directly from data
- Theorem. If $n,k\to\infty$, $k/n\to 0$, then $\hat{f}(x)\to$ E(Y | X=x)

Algorithms for Predictive Modelling – Contents

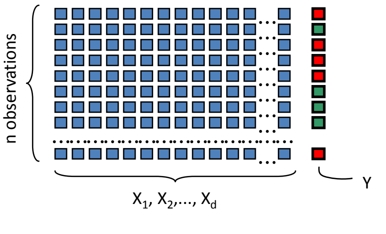
- Regression
- **Classification**
- Auxiliary topics:
  - Estimation of prediction error
  - Reduction of dimensionality

Algorithms for building classifiers

- Linear, nonlinear discriminant analysis
- Logistic regression
- Naive Bayes
- Classification and regression trees
- Perceptron algorithm, neural networks
- Support vector machines
- Nonparametric classifiers (k nearest neighbours)
- …

Classification

- Notation
  $X \in R^d$ – input variables (random variables)
  $Y \in C=\{c_1, c_2,\dots,c_M\}$ – possible values of Y
  $x \in c_i$ if $Y(x)=c_i$ (x *belongs to* class c_i)

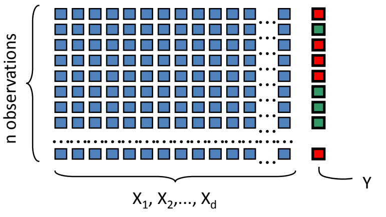


- We again assume that the joint probability distribution Pr(X,Y) of X and Y is known

Classification

- *Loss matrix* L_{MxM} (to punish misclassification decisions of classifier f)

  $L(i,j)=0$ for $x \in c_i$ and $f(x)=c_j$ and $i=j$
  $L(i,j)=1$ for $x \in c_i$ and $f(x)=c_j$ and $i \neq j$



- **We look for classifer f(X) minimizing the *expected prediction error*:**

$$EPE = E\big(L\big(Y, f(X)\big)\big)$$

61

Bayes Classifier

- Solution – *Bayes classifier*

$$f(x) = c_k \quad \text{for} \quad \Pr(c_k \mid X = x) = \max_{c \in C} \Pr\big(c \mid X = x\big)$$

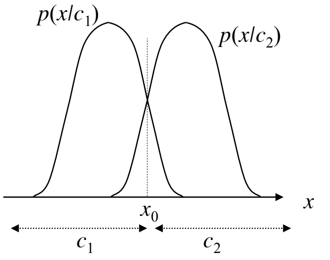
- Rule: classify x into class c_k which proves *most likely* under conditional distribution $\Pr(Y|X=x)$
- The Bayes classifier is <u>optimal</u> with respect to minimizing the classification error probability

62

Bayes Classifier

- Bayes classifier – simple rule:
  $Pr(c_1|x) > Pr(c_2|x)$ → classify x to c_1
  $Pr(c_1|x) < Pr(c_2|x)$ → classify x to c_2
- Problem: estimation of $Pr(c_i|x)$ *from data* is hard, easier to estimate $Pr(x|c_i)$
- Bayes rule:

$$\Pr(c_i \mid x) = \frac{\Pr(x \mid c_i)\Pr(c_i)}{\Pr(x)}$$



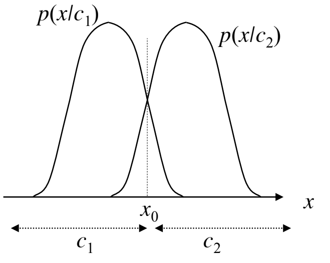
63

Bayes Classifier

- Bayes rule:

$$\Pr(c_i \mid x) = \frac{\Pr(x \mid c_i)\Pr(c_i)}{\Pr(x)}$$

- $Pr(c_i|x)$ – known as *a posteriori* probabilities
- $Pr(c_i)$ – *a priori* probabilities, can be estimated from training data (as frequencies of samples of various classes)



64

Bayes Classifier

- Bayes classifier – equivalent form:

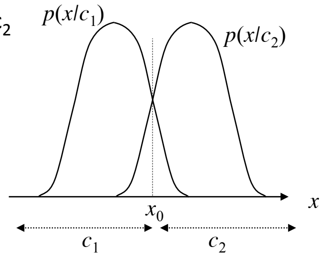
  $Pr(x|c_1)Pr(c_1) > Pr(x|c_2)Pr(c_2)$ → classify x to c_1

  $Pr(x|c_1)Pr(c_1) < Pr(x|c_2)Pr(c_2)$ → classify x to c_2

- Or, for *equiprobable* classes:

  $Pr(x|c_1) > Pr(x|c_2)$ → classify x to c_1

  $Pr(x|c_1) < Pr(x|c_2)$ → classify x to c_2



65

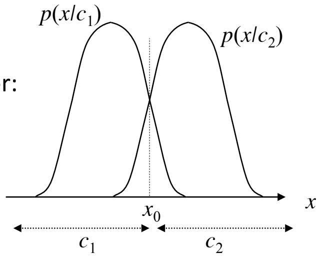
Bayes Error

- Rule for equiprobable classes:

  $Pr(x|c_1) > Pr(x|c_2)$ → classify x to c_1

  $Pr(x|c_1) < Pr(x|c_2)$ → classify x to c_2

- Total probability of committing classification error:



$$P_e = \Pr(c_2)\int_{-\infty}^{x_0} p(x \mid c_2)dx + \Pr(c_1)\int_{x_0}^{+\infty} p(x \mid c_1)dx$$

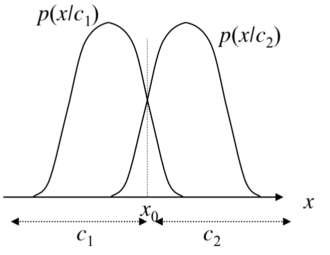
$$P_e = \frac{1}{2}\left(\int_{-\infty}^{x_0} p(x \mid c_2)dx + \int_{x_0}^{+\infty} p(x \mid c_1)dx\right)$$

(For equiprobable classes)

66

Bayes Error

- Bayes error – error due to *overlapping* features
- The only way to reduce this error is to provide more information about items classified (more features)



Discriminant functions

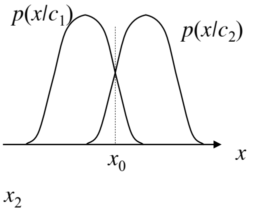
- Given Bayes rule:
  $Pr(c_1|x) > Pr(c_2|x)$ → classify x to c_1
  $Pr(c_1|x) < Pr(c_2|x)$ → classify x to c_2

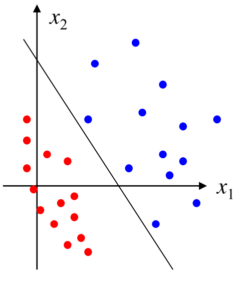
  partitions the feature space into disjoint regions. *Decision surface* in the boundary between contiguous regions:

  $g(x) = Pr(c_1|x) – Pr(c_2|x) = 0$

- Then x is classified as:

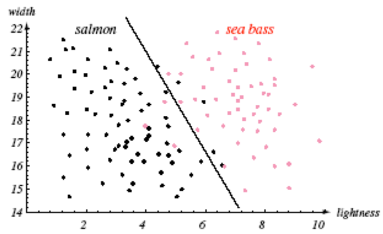
  c = sign(g(x))





Linear Methods in Classification

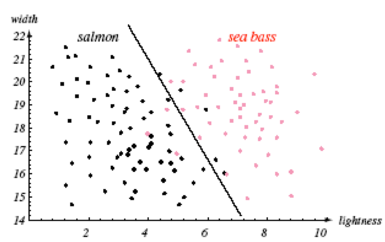
- How to estimate the *decision hyperplanes* given training data ?
  - Analytical solution for $p(x|c_i)$, i=1,…,M *normally* distributed
    - LDA
    - QDA
  - Iterative algorithms to fit separating hyperplanes for *non-normally* distributed data



Linear Discriminant Analysis

- LDA Linear Discriminant Analysis

$$c_i = sign\left(\sum_{j=1}^{d} \beta_j X_j + \beta_0\right)$$

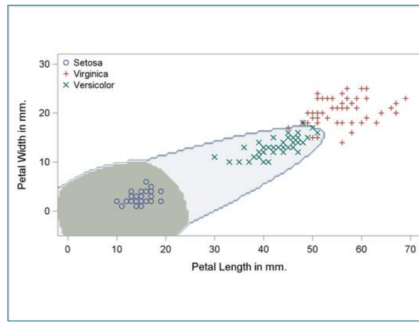


- LDA defines the Bayes (optimal) classifier for $p(x|c_i)$ normally distributed (with the same covariance matrix in all classes)
- Then $\beta_0, \beta_1, \ldots, \beta_d$ are computed based on means and covariances of features in different classes

Quadratic Discriminant Analysis

- QDA Quadratic Discriminant Analysis

$$c_i = sign\left(\beta_0 + \sum_{j=1}^{d}\beta_j X_j + \sum_{j=1}^{d}\sum_{k=1}^{d}\beta_{jk} X_j X_k\right)$$



- QDA defines the Bayes (optimal) classifier for the general case of $p(x|c_i)$ normally distributed

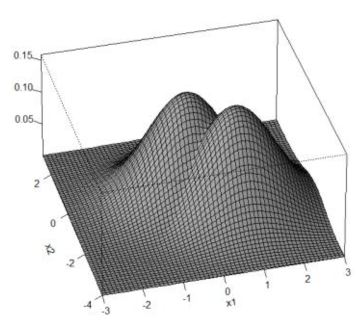
71

Naive Bayes Classifier

- Problem: in high dimensional data $x=(x_1,\ldots,x_d)$ estimation of class-conditional densities $p(x \mid c_j)$, $j=1,\ldots,m$ is difficult (*curse of dimensionality*)

- Naive Bayes is based on the assumption:

$$p(x|c_j) = \prod_{k=1}^{d} p(x_k|c_j)$$

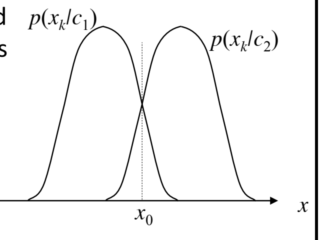


i.e. we assume that in each class j, the features are independent

72

Naive Bayes Classifier

- Estimation of one-dimensional marginal densities $p(x_k \mid c_j)$ – easy:
  - Continuous features: $p(x_k \mid c_j)$ can be estimated using one-dimensional kernel density estimates or univariate normal distribution
  - Discrete features: $p(x_k \mid c_j)$ based on the observed frequency (histogram-based estimates)



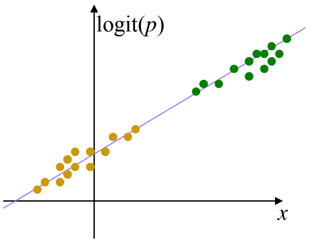
- Easy to mix different feature types in feature vector
- In reality features *are* dependent, depite this NB often performs remarkably well…

73

Logistic Regression

- Idea of logistic regression
  - Y – discrete, e.g., with values 0 or 1
  - p=Pr(Y=1|X=x)
  - Logit(p)=log(p/(1-p))
  - We model logit(p) as a linear function of features

$$\mathrm{logit}\,(p) = \sum_{j=1}^{d} \beta_j X_j + \beta_0$$



74

Linear Methods in Classification – Separating Hyperplanes

- Problem: given *linearly separable* training data, find a separating hyperplane (defined by $\boldsymbol{\beta}$, β_0):

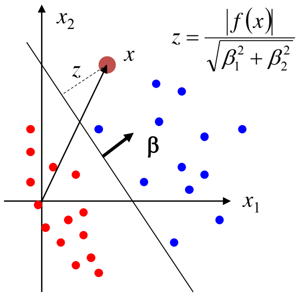
$$f(\mathbf{x}) = \sum_{j=1}^{d} \beta_j X_j + \beta_0 = 0$$

$$f(\mathbf{x}) = \langle \boldsymbol{\beta}, \mathbf{x} \rangle + \beta_0 = 0$$

- Class c_i of x is then found as:

$$c_i = sign(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + \beta_0)$$

β=[β_1, β_2]

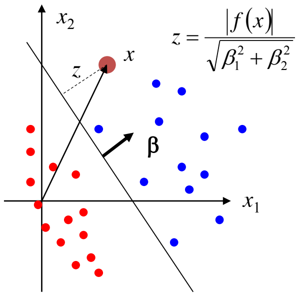


75

Linear Methods in Classification – Separating Hyperplanes

- Here no assumption is made regarding distribution of training data Pr(x|c_i)
- ? How to find $\boldsymbol{\beta}$, β_0 → perceptron algorithm

β=[β_1, β_2]



76

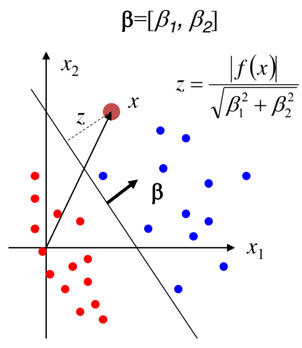
Linear Methods in Classification – Perceptron Algorithm

- Idea: find $\boldsymbol{\beta}$ by solving an optimization task – minimize cost J($\boldsymbol{\beta}$) of misclassified items:

$$J(\boldsymbol{\beta}) = \sum_{misclassifed\ \mathbf{x}} \left| \langle \boldsymbol{\beta}, \mathbf{x} \rangle \right|$$

- $\boldsymbol{\beta}$ found by iterative minimization of J($\boldsymbol{\beta}$) – gradient descent method

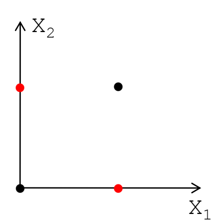
$$\boldsymbol{\beta}(t+1) = \boldsymbol{\beta}(t) - \rho_t \frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$



Perceptron for Non-linearly Separable Data ?

- Example of non-linearly separable data:

X_1	X_2	Y
0	0	**A**
0	1	**B**
1	0	**B**
1	1	**A**

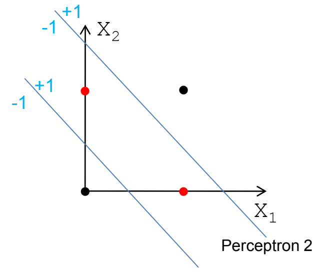


- However, after transformation of variables, problem becomes linearly separable…

Non-linearly separable data

- Mapping based on output of two perceptrons

X_1	X_2	Y	P_1	P_2
0	0	**A**	-1	-1
0	1	**B**	+1	-1
1	0	**B**	+1	-1
1	1	**A**	+1	+1



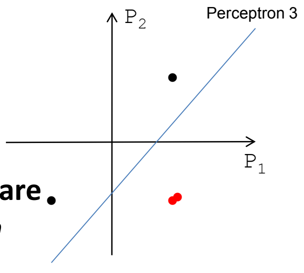
79

Non-linearly separable data

- Mapping based on output of two perceptrons

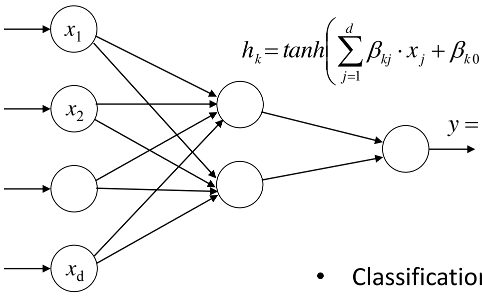
X_1	X_2	Y	P_1	P_2
0	0	**A**	-1	-1
0	1	**B**	+1	-1
1	0	**B**	+1	-1
1	1	**A**	+1	+1

- In new coordinate system, the two classes **are** linearly separable → *multilayer perceptron*



80

Neural Nets: Multilayer Perceptron



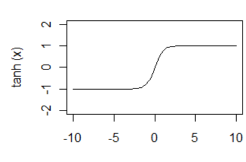
$$h_k = tanh\left(\sum_{j=1}^{d} \beta_{kj} \cdot x_j + \beta_{k0}\right)$$

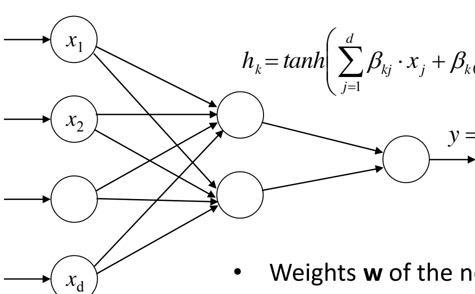
$$y = tanh\left(\sum_{k=1}^{nHID} \alpha_k \cdot h_k + \alpha_0\right)$$

- Classification rule for x:
  f(x) = sign (y)

81

Neural Nets: Multilayer Perceptron





$$h_k = tanh\left(\sum_{j=1}^{d} \beta_{kj} \cdot x_j + \beta_{k0}\right)$$

$$y = tanh\left(\sum_{k=1}^{nHID} \alpha_k \cdot h_k + \alpha_0\right)$$

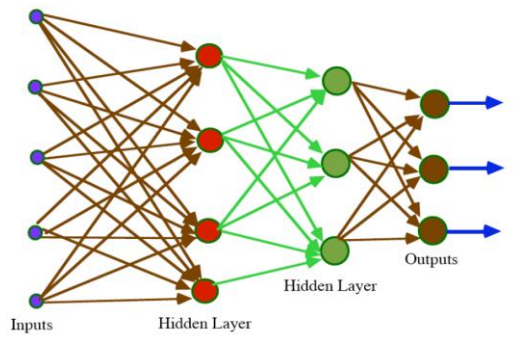
- Weights **w** of the network (parameters α,β) fitted by the backpropagation algorithm

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \rho_t \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$$

82

Neural Nets

Output can involve more neurons (non-binary classification)



83

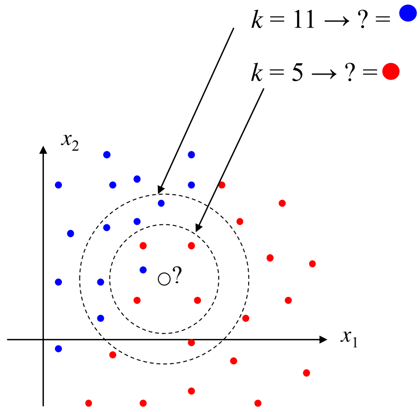
Neural Nets

- Interval (quantitative) inputs: one input neuron per input variable
- Ordinal or nominal inputs: variable with N levels is changed to N dummy variables (with values of 0,1 used to code class membership)
- Drawback: NN is a „Black-Box" tool – impossible to understand prediction process

84

K-Nearest Neighbours

- Idea: predict target for new set of inputs based of *k* nearest neighbours in training data set
- Example of a *nonparametric* approach
- Can be regarded as an approach to compare $p(x|c_1)$ vs $p(x|c_2)$ – directly from data



K-Nearest Neighbours

- E.g., voting records for US congressmen

n y n y y y n n n y ? y y y n y --> republican
n y n y y y n n n n n y y y n ? --> republican
? y y ? y y n n n n y n y y n n --> democrat
n y y n ? y n n n n y n y n n y --> democrat
y y y n y y n n n n y ? y y y y --> democrat
n y y n y y n n n n n n y y y y --> democrat
n y n y y y n n n n n n ? y y y --> democrat
n y n y y y n n n n n n y y ? y --> republican
n y n y y y n n n n n y y y n y --> republican
y y y n n n y y y n n n n n ? ? --> democrat
n y n y y n n n n n ? ? y y n n --> republican
n y n y y y n n n n y ? y y ? ? --> republican

- Prediction:

y n n y y n y y y n n y y y n y --> ?
y n n y y n y y y n n y y y n y --> republican

Note: subsequent 'n', 'y' denote *against* or *in favour of* vote in following matters:

1. more-support-for-handicapped-infants
2. water-project-cost-sharing
3. adoption-of-the-budget-resolution
4. physician-fee-freeze
5. el-salvador-aid
6. religious-groups-in-schools
7. anti-satellite-test-ban
8. aid-to-nicaraguan-contras
9. mx-missile
10. immigration
11. synfuels-corporation-cutback
12. education-spending
13. superfund-right-to-sue
14. crime
15. duty-free-exports
16. export-administration-act-south-africa

K-Nearest Neighbours

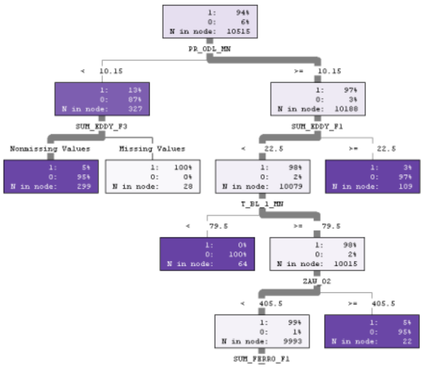
- We need to define how to measure distance between points:
  - Euclidean distance
  - City-block (Manhattan distance) - sum of the absolute differences between corresponding values
  - Correlation distance = 1-cor(v_1, v_2)
    (cor – correlation coefficient between vectors v_1, v_2)

87

Decision Trees

- Root: whole data set
- Recursively partition data to increase *purity* of partitions
- Leaves – correspond to classification decisions

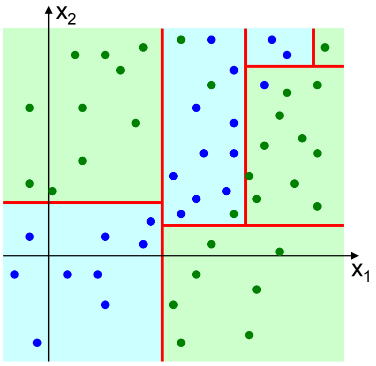
- Measures of purity: entropy, Gini-index, …
- Partitioning continues until partitions become pure or small
- Binary trees – most frequently used
- Algorithms CART, C4.5, C5.0



88

Decision Trees

- Decision process easy to understand
- Drawback: poor stability (small changes in data may lead to different trees)
- Hence: *random forests* (L. Breiman)

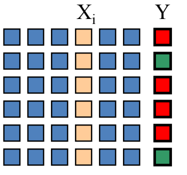


Algorithms for Building Decision Trees – Problems

- How to select an attribute to use for a data split?
- How long should we follow with the process of splitting?
- What is the *right* height/size of the tree?

Selection of Attribute for Next Split

- **Approach I**: Based on some measure of *relationship* of variables X_i and Y
  - Idea: for each X_i i=1,2,…d, compute pValue of test for H_0: X_i and Y independent
  - Split node S using variable X_i, for which pValue is smallest (providing pValue < 0.05 (possibly with multiple testing correction))
  - Possible tests: Chi-square, F or ANOVA

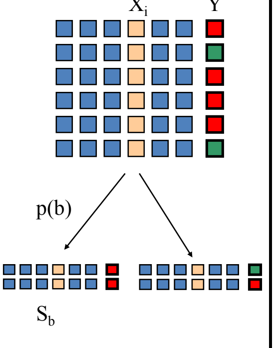


Selection of Attribute for Next Split

- **Approach II**: Maximize some criterion of *quality of split* (e.g. entropy reduction, variance reduction etc.)
- Idea:
  - Define I(S) – measure of node's S *impurity*
  - Considering split based on variable A, define measure of quality of this split:

    $$\mathbf{W(S,A) = I(S) - \sum p(b) I(S_b)}$$

    (sum over all branches b leaving node S
    p(b) – probability of selecting branch b)
  - Select split according to A, maximizing the measure W



Selection of Attribute for Next Split – definitions of I(S)

- Entropy $(Y \in C=\{c_1, c_2, \ldots, c_M\})$

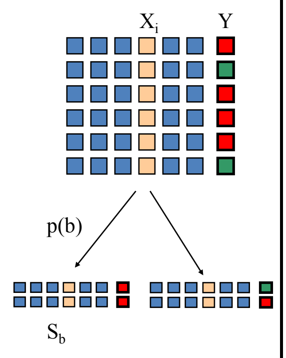
$$Entropy(S) = -\sum_{i=1}^{M} p_i \log_2 p_i$$

- Gini index

$$Gini(S) = 1 - \sum_{i=1}^{M} \left(\frac{n_i}{|S|} \right)^2$$

- Both measures realize:
  – max value for equal probabilities of all classes (=1/M)
  – 0 if p_i=1 for some i (all data belong to a single class)

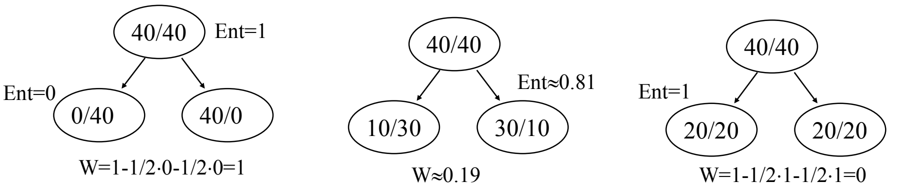
p_i – % observations of class i in node S
n_i – number of obs of class i in S



Selection of Attribute for Next Split

- Consider 3 competing splits of a node
- Notation: n_1/n_2 – number of observations of class „1" / „2" in node
- We select the split maximizing W:

$$W(S,A) = Entropy(S) - \sum_{\text{branch } b} \frac{|S_b|}{|S|} Entropy(S_b)$$



Algorithms for Building Decision Trees – Problems

✓ How to select an attribute to use for a data split?

- How long should we follow with the process of splitting?
  - Rule: stop if node pure or small (min n – parametr of algorithm)
- What is the *right* height/size of the tree?

Algorithms for Building Decision Trees – Problems

✓ How to select an attribute to use for a data split?

✓ How long should we follow with the process of splitting?

- What is the *right* height/size of the tree?
  - Problem:
    find the balance between *overfitting* vs *oversimplifying*
    (too complex vs too simple tree)
  - Rule: estimate error (misclassification cost) for train and test data
  - Prune higher tree to decrease the cost (measured for test data)

Trees *vs.* Neural Nets

- Trees:
  - Classification process can be understood (and expressed as set of simple rules). Classification rules give insight into data
  - Can be unstable (small change in input data may change structure of tree) → random forests…
- Neural nets:
  - 'Black Box' – classification process cannot be understood
  - More stable than trees

97

Optimal separating hyperplanes

- Linear classifier f (for ($y_i \in$ {-1,+1}, i=1,2,…,N)):

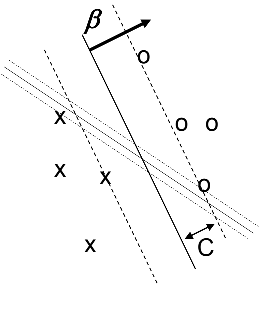
  $C(x_i)$=sign $f(x_i)$

- Task
  - Find separating hyperplane with the constraint: maximize the *separating margin* → solve optimization problem:

$$\max_{\beta,\beta_0} C$$

$$subject\ \ to\ \ y_i\left(\beta_0 + \langle \beta, x_i \rangle\right) \geq C,\ \ i = 1,2,\ldots,N$$

$$f(x) = \beta_0 + \langle \beta, x \rangle = 0$$



98

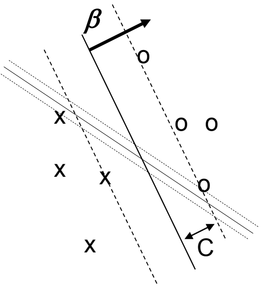
Optimal separating hyperplanes

- Solution:

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$$

with $\alpha_i \neq 0$ only for points on the edge of the separating slab (**support vectors**)

$$f(x) = \beta_0 + \langle \beta, x \rangle = 0$$



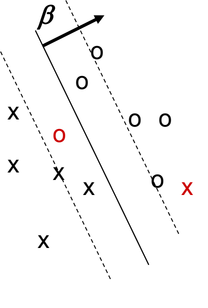
99

Support vector classifier

- Allow for *non-linearly separable* data
- Find separating hyperplane→ solve the same optimization problem with different constraints:

$$\max_{\beta,\beta_0} C$$
$$subject\ \ to\ \ y_i(\beta_0 + \langle \beta, x_i \rangle) \geq C(1-\xi_i),\ \ i = 1,2,\ldots,N$$
$$with \qquad \sum_i \xi_i \leq const$$

$$f(x) = \beta_0 + \langle \beta, x \rangle = 0$$



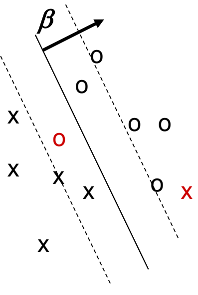
100

Support vector classifier

- Values ξ_i indicate proportional amout by which prediction $\beta_0+\langle\beta,x_i\rangle$ is on the wrong side of the margin
- We bound $\sum\xi_i$
- Solution: β depends on support vectors only

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$$

$$f(x) = \beta_0 + \langle \beta, x \rangle = 0$$



Support vector machine

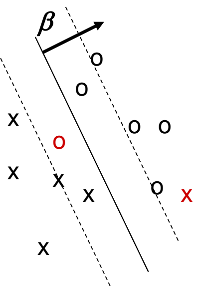
- Putting solution for β into definition of f(x)

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \langle x_i, x \rangle + \beta_0$$

- Dual representation: data appears in dot products only
- This cannot deal with non-linearly separable (or noisy) data
-
  Solution: Move data to higher dimensionality feature space and separate data there

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \langle h(x_i), h(x) \rangle + \beta_0$$

$$f(x) = \beta_0 + \langle \beta, x \rangle = 0$$



Support vector machine

- We do not need the transformation h() – we need only *kernels* !

$$K(x,x') = \langle h(x), h(x') \rangle$$

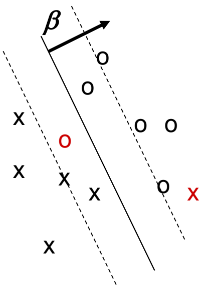
- Possible kernels

$K(X,X') = \langle X, X' \rangle$ linear

$K(X,X') = (1 + \langle X, X' \rangle)^2$ polynomial

$K(X,X') = \exp(\sigma \|X - X'\|^2)$ radial-basis

$$f(x) = \beta_0 + \langle \beta, x \rangle = 0$$



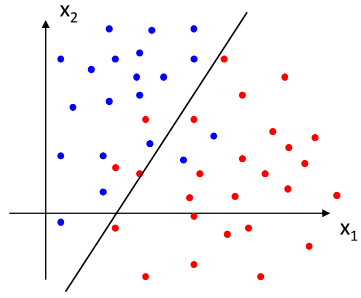
103

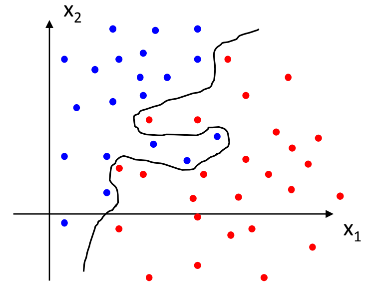
Auxiliary Topics

- Classifier complexity
- Estimation of prediction error
- „Curse of dimensionality”
- Feature selection
- PCA

104

Classifier Complexity and Risk of Overfitting

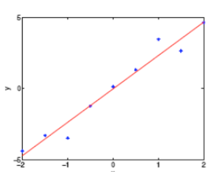




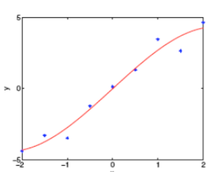
Too complex classifier looses *generalization* property;
Becomes prone to *overfitting*

105

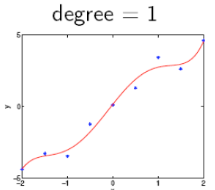
Complexity of Regression Models



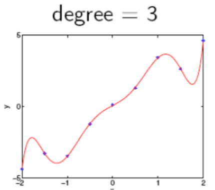
degree = 1



degree = 3



degree = 5



degree = 7

Too complex models
may overfit data

106

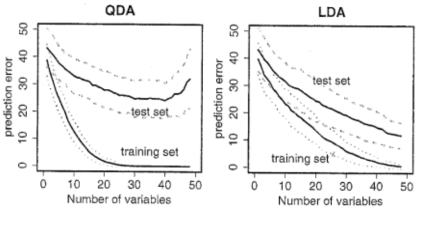
Classifier Overfitting

- Training a classifier – attempt to fine-tune classifier parameters to minimize the *training error*
- Performance of classifier is to be verified on independent (new) test data
- Often a large gap observed between *training error* and *test error*
- This problem is called **overfitting**

107

Overfitting and Classifier Complexity

- Overfitting increases with:
  - Model complexity
  - Dimensionality of the problem
- Example (Markowetz, Methods Inf Med. 3/2005) LDA vs QDA with changing number of variables

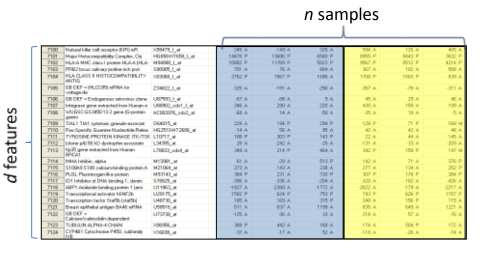


- Too simple classifer – not able to grasp the relationship between target and input parameters
- Too complex – overfitting → trade-off needs to be sought between *flexibility* of classifier and its *ability to generalize*

108

Classification Based on High Throughput Experiments Data

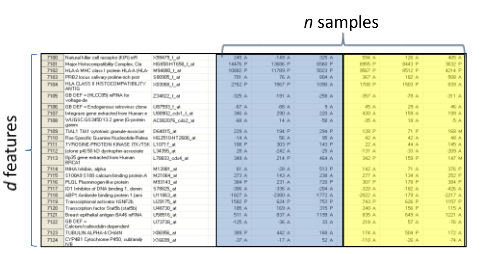
- High throughput data: d>>n
- **Interesting fact: in *p* dimensions a simple linear classifier can always separate *p*+1 points**
- Hence, it is always possible to perfectly separate such training data set → Overfitting as the major problem



109

Classification Based on High Throughput Experiments - Problems

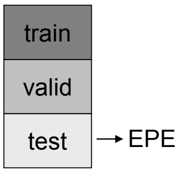
- Reduction of dimensionality important to avoid overfitting
  - Feature selection based on statistical hypothesis tests (discriminatory capability of features tested independently)
  - PCA
- Small size of available data (n)
  - Insufficient data to train the model and test it on *indepentdent*
  - *Data reuse* methods

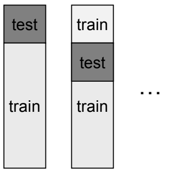


110

Estimation of Expected Prediction Error – Data Reuse Methods

- „Standard" approach (n samples >> d dimension)
  - Partition data into *train/validation/test* subsets
  - Validation – used to fine-tune model's parameters (nHiddenNeurons, nIterations, nInputs, SVM *C* and γ, …)
  - Test – to estimate EPE
- n<<d case → no data to *validate* and *test*
- Data reuse to estimate EPE, e.g.,
  - K-fold cross validation
  - Leave-out-one cross validation, etc.





111

Performance of Classifiers

Confusion matrix

Predicted

Actual		Class 1	Class 2
	Class 1	**a**	**b**
	Class 2	**c**	**d**

Sensitivity = a/(a+b)

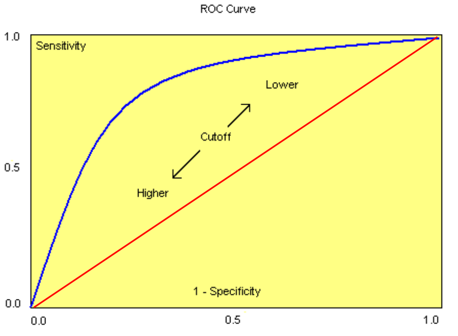
Specificity = d/(c+d)

112

Performance of Classifiers

- Receiver operating characteristic (ROC)
- Each point represents a cut-off probability

- Area under ROC:
  0.5 – worthless model
  1 – perfect classifier



„Curse of Dimensionality"

- Example: N points in p dimensions distributed uniformly in a unit ball. Then the median distance from the origin to the closest point is:

$$d=(1-0.5^{1/N})^{1/p}$$

p	d
1	0.00069
2	0.02632
3	0.08849
10	0.48313
50	0.86460

- „Curse of dimensionality" – in high dimensions training data sparsly populate input space
- This limits, in high dimensionality problems, practical use of various data mining techniques such as
  - Clustering
  - K-nearest neighbour, etc.

Principal Components Analysis - Idea

- Transform the original set of *p* variables (on *n* observations) to uncorrelated set of p *latent* variables – *principal components*
- Variability in data is captured in first few latent variables (ordered in the descending order of explained variability)
- Applications
  - Technique of analysis of high dimensional data / data summarization / explanatory data analysis
  - First principal components are used to reduce the number of variables in predictive modelling, clustering, etc.

115

PCA – Technical Details

- Given a data set with p numeric variables, you can compute p principal components.
- Each principal component is a linear combination of the original variables, with coefficients equal to the eigenvectors of the correlation or covariance matrix.
- The eigenvectors are customarily taken with unit length.
- The principal components are sorted by descending order of the eigenvalues, which are equal to the variances of the components.

116

Example: PCA for Information Compression

- Analysis of job performance of policemen (as seen by their bosses)
- Each policeman rated in 14 categories (1=fail, 10=outstanding)

	Communication Skills	Problem Solving	Learning Ability	Judgment Under Pressure	Observation Skills	Willingness to Confront Problems	Interest in People	Interpersonal Sensitivity	Desire for Self-Improvemen	Appearance	Dependability	Physical Ability	Integrity	Overall Rating
1	2	6	8	3	8	8	5	3	8	7	9	8	6	7
2	7	4	7	5	8	8	7	6	8	5	7	6	6	7
3	5	6	7	5	7	8	6	3	7	7	5	8	7	5
4	6	7	8	6	9	7	7	7	9	8	8	9	9	7
5	9	9	8	9	9	8	9	9	8	9	9	8	9	9
6	7	4	7	5	8	8	7	6	8	5	7	6	6	7
7	5	6	7	5	7	8	6	3	7	7	5	8	7	5
8	6	7	8	6	9	7	7	7	9	8	8	9	9	7
9	9	9	8	9	9	8	9	9	8	9	9	8	9	9
10	7	4	7	5	8	8	7	6	8	5	7	6	6	7
11	5	6	7	5	7	8	6	3	7	7	5	8	7	5
12	6	7	8	6	9	7	7	7	9	8	8	9	9	7
13	9	9	8	9	9	8	9	9	8	9	9	8	9	9

Example – cnt'd

Correlation matrix (part)

	Communication Skills	Problem Solving	Learning Ability	Ju U Pr
Communication Skills	1.0000	0.6280	0.5546	
Problem Solving	0.6280	1.0000	0.5690	
Learning Ability	0.5546	0.5690	1.0000	
Judgment Under Pressure	0.5538	0.6195	0.4892	
Observational Skills	0.5381	0.4284	0.6230	
Willingness to Confront Problems	0.5265	0.5015	0.5245	
Interest in People	0.4391	0.3972	0.2735	

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	6.54740242	4.77468744	0.5036	0.5036
2	1.77271499	0.76747933	0.1364	0.6400
3	1.00523565	0.26209665	0.0773	0.7173
4	0.74313901	0.06479499	0.0572	0.7745
5	0.67834402	0.22696368	0.0522	0.8267
6	0.45138034	0.06922167	0.0347	0.8614
7	0.38215866	0.08432613	0.0294	0.8908
8	0.29783254	0.02340663	0.0229	0.9137
9	0.27442591	0.01208809	[illegible]	[illegible]
10	0.26233782	0.01778332	0.0202	0.9550
11	0.24455450	0.04677622	[illegible]	[illegible]
12	0.19777828	0.05508241	0.0152	0.9890
13	0.14269586		0.0110	1.0000

Eigenvalues sum up to total variance

First component explains 50% of variation

First 5 components explain ~ 83%

	Prin1	Prin2	Prin3	Prin4
Communication Skills	0.303548	0.052039	-.329181	-.227039
Problem Solving	0.278034	0.057046	-.400112	0.300476
Learning Ability	0.266521	0.288152	-.354591	-.020735
Judgment Under Pressure	0.294376	-.199458	-.255164	0.397306
Observational Skills	0.276641	0.366979	0.065959	0.035711
Willingness to Confront Problems	0.267580	0.392989	0.098723	0.184409
Interest in People	0.278060	-.432916	0.118113	0.046047
Interpersonal Sensitivity	0.253814	-.495662	-.064547	-.060000
Desire for Self-Improvement	0.299833	0.099077	0.061097	-.211279
Appearance	0.237358	0.190065	0.248353	-.544587
Dependability	0.319480	-.049742	0.169476	-.156070
Physical Ability	0.213868	0.097499	0.614959	0.514519
Integrity	0.298246	-.301812	0.190222	-.169062

PRIN1, PRIN2, … -- Eigenvectors of the correlation matrix

PC1 = linear combination of the original variables with coefficients given in the first eigenvector

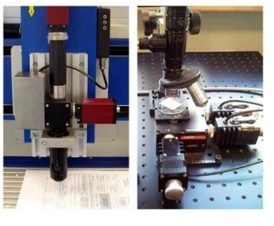
PC1=0.303548*(Communication_skills)
+0.278034*(Problem solving)+…

PC1 ~ overall performance
PC2 ~ smart, not socialized
PC3 ~ strong, not very brilliant

119

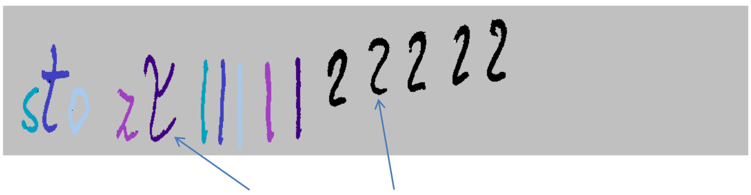
Example: PCA for ink recognition based on spectral data

- Problem: we want to verify if two signatures have been written with the same ink
- Signatures (images) scanned with the spectral scanner



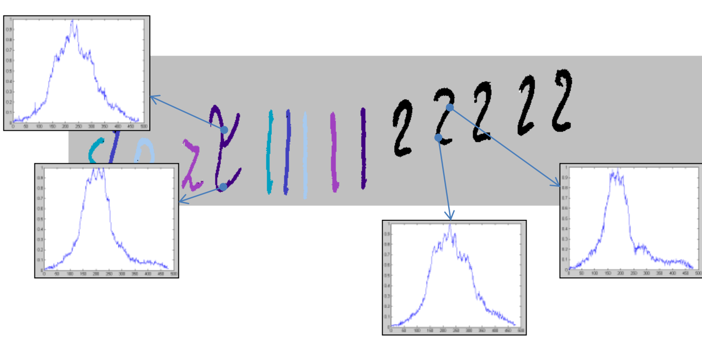
Project details: http://skaner.mvlab.pl

Example: PCA for ink recognition based on spectral data



- Task: verify the H0 that two signatures have been written with the same ink

Example: PCA for ink recognition based on spectral data

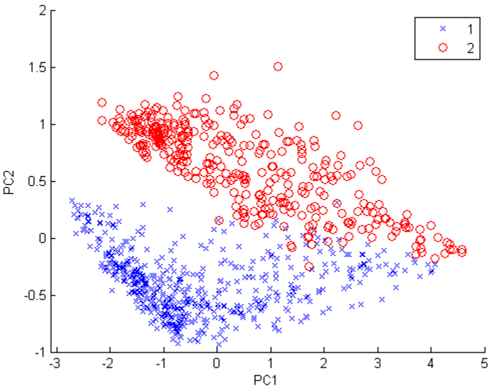


```
[480x626  single]
[480x1228 single]
[480x903  single]
[480x708  single]
[480x1248 single]
[480x653  single]
[480x671  single]
[480x696  single]
[480x819  single]
[480x705  single]
[480x1126 single]
[480x862  single]
[480x1033 single]
[480x999  single]
[480x1106 single]
```

- Every point in the signature is represented by a vector of spectral data (e.g. 480 spectral values per point)
- PCA is useful for reduction of dimensionality of the spectral data

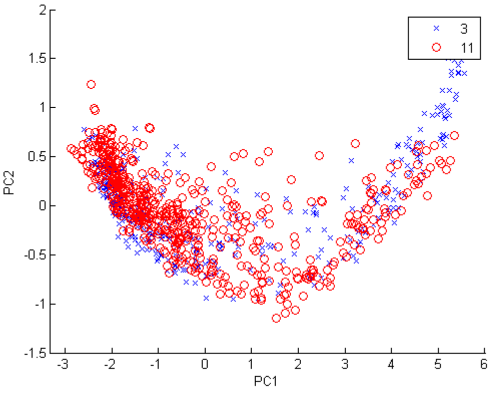
Example: PCA for ink recognition based on spectral data

Visualization of the compared inks in the space of the first two components: different inks

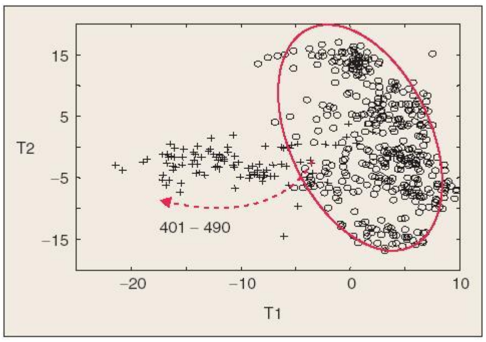


Example: PCA for ink recognition based on spectral data

Visualization of the compared inks in the space of the first two components: the same ink used to write two samples of text



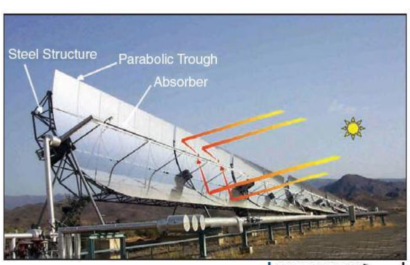
Example: PCA for process control



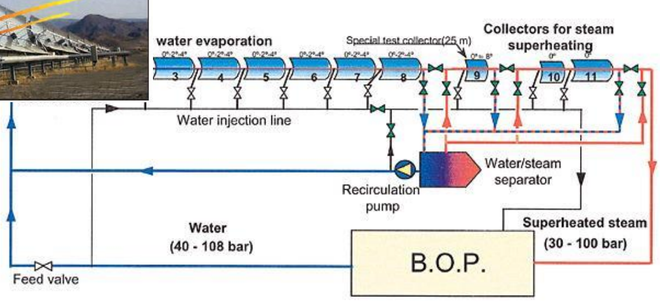
Idea
Process variables transformed into set of uncorrelated *latent variables*:
- PCA method
- PLS method

Scatter plot of first two latent variables used to determine area of:
- Proper system operation
- Abnormal conditions

Example: PCA for Process Control – DISS Solar Plant



- www.psa.es

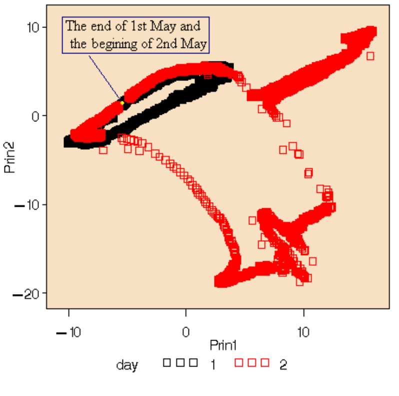


Example: DISS Monitoring Data

- DISS monitoring data
  - 55MB data / day, interval 5 sec.
  - ~600 variables measured / recorded (flows, temperatures, pressures of the solar field, meteorological data, monitoring of power block BOP, etc.
- Purpose of analysis is to detect / predict faults
  Fault – temperature difference in a cross section surpasses 50^{o}C (collector goes to a desteer state)

127

Example: System State Trajectory

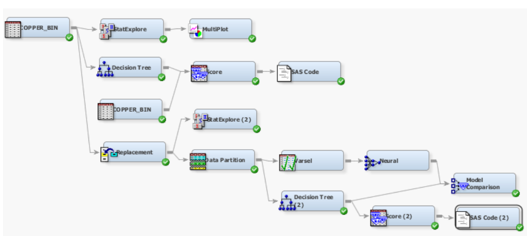


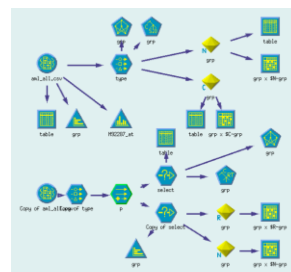
- Idea: PCA model: scatter plot of first two principal components
- Day 1 – recirculation and feed pumps not running
- Day 2 – correct plant operation

128

Towards Standarization of DM Tools (PMML)

- Problem: how deploy models created with different DM tools into a production DBMS scoring system?





Predictive Modeling Markup Language (PMML)

- XML based language to express data mining models
- Purpose: interchange DM models between DM / BI tools (e.g., predictive model built in one tool to be used for scoring in another tool)
- Developed by Data Mining Group (IBM, Oracle, Microsoft, Microstrategy, prudsys, SAS, SPSS, StatSoft,…) www.dmg.org
- Specification available at http://sourceforge.net/projects/pmml (here XML Schema pmml-3-1.xsd)

DM Model Expressed in PMML

- PMML document describing a DM model is an XML document based on PMML XML schema (pmml-3-1.xsd)
- PMML standard (ver. 3-1 allows to describe models:
  * Association Rules
  * Decision Trees
  * Center-Based & Distribution-Based Clustering
  * Regression
  * General Regression
  * Neural Networks
  * Naive Bayes
  * Sequences
  * Text
  * Ruleset
  * Support Vector Machine

Example: Decision Tree in PMML

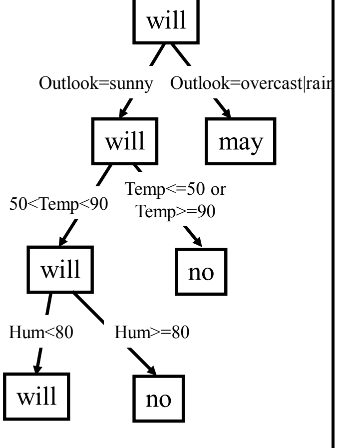
- Build a decission tree classifier to decide whether to play golf (famous problem in DM ☺)
- Training data

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

Example: Decision Tree in PMML

- Node has:
  - One predicate (defines condition to select this node),
  - Child nodes (if not leaf node; no child nodes → leaf).
- Scoring procedure (given observation X):
  - Start with root node (predicate=True)
  - Check if predicate of first child is True for X.
    If YES: move to this node. Repeat procedure for child nodes of this node (if leaf: return Node score)
  - If NO: return to root and repeat procedure for next child

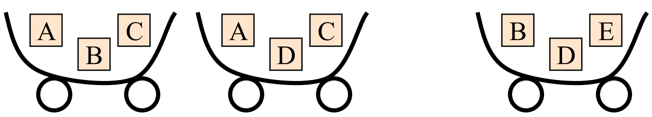

```
  <?xml version="1.0" ?>
- <PMML version="3.1">
    <Header copyright="www.dmg.org" description="A very small binary tree model to sh
      structure." />
  + <DataDictionary numberOfFields="5">
  - <TreeModel modelName="golfing" functionName="classification">
    + <MiningSchema>
    - <Node score="will play">
        <True />
      - <Node score="will play">
          <SimplePredicate field="outlook" operator="equal" value="sunny" />
        - <Node score="will play">
          - <CompoundPredicate booleanOperator="and">
              <SimplePredicate field="temperature" operator="lessThan" value="90" />
              <SimplePredicate field="temperature" operator="greaterThan" value="50" />
            </CompoundPredicate>
          - <Node score="will play">
              <SimplePredicate field="humidity" operator="lessThan" value="80" />
            </Node>
          - <Node score="no play">
              <SimplePredicate field="humidity" operator="greaterOrEqual" value="80" />
            </Node>
          </Node>
        + <Node score="no play">
        </Node>
      - <Node score="may play">
        - <CompoundPredicate booleanOperator="or">
            <SimplePredicate field="outlook" operator="equal" value="overcast" />
            <SimplePredicate field="outlook" operator="equal" value="rain" />
          </CompoundPredicate>
        + <Node score="may play">
        + <Node score="no play">
        </Node>
      </Node>
    </TreeModel>
  </PMML>
```



Association Rules

- Problem: given a transaction database, find *all* association rules A→B with given min *support* and min *confidence*

- Algorithms should be efficient for large databases, e.g., :
  - >10^5 items
  - >10^6 transactions

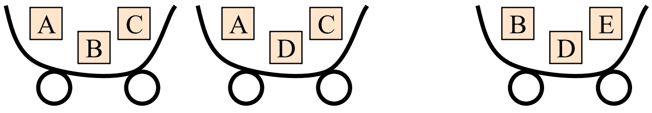


135

Important Concepts

- **Support** of rule = frequency of {A,B} in transaction database
- **Confidence** of rule = Pr(B|A)

- Another interesting parameter of a rule is **lift** = Pr(B|A) / Pr(B)
  Pr(B)=prob. of buying B, no rule known
  Interesting rules significantly *lift* probability of buying B



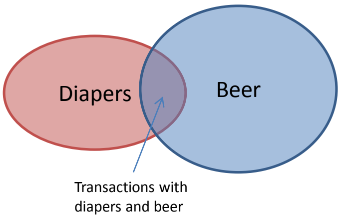
136

Example

- Rule: "If a customer purchases diapers, then 40% of the times he/she will purchase beer"

All transactions:	1.000.000
Trans. with diapers:	50.000
Trans. with beer:	200.000
Trans. with diapers and beer:	20.000

Support = 2%
Confidence(D→B) = 20K/50K = 40%
Pr(B) = 20%   Lift = 2
Confidence(B→D) = ?



Algorithms for Association Rules – Notation

$\mathbf{I}=\{i_1,i_2,\ldots,i_m\}$	set of items
$\mathbf{D}$	set of transactions
$T\subseteq\mathbf{I}$	transaction
$X\subseteq\mathbf{I}$, $Y\subseteq\mathbf{I}$	itemsets

- Transaction T contains itemset X if $X\subseteq T$
  $\mathbf{T_X}$ - set of such transactions
- Support sup(X) of *itemset* X: percent of transactions in **D** containing X, $\mathrm{sup}(X)=|\mathbf{T_X}|/|\mathbf{D}|$

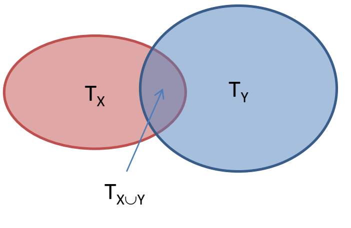
Algorithms for Association Rules – Notation

- Association rule is an implication $X \Rightarrow Y$, where $X \subset \mathbf{I}$, $Y \subset \mathbf{I}$ and $X \cap Y = \varnothing$
- *Confidence* c of rule $X \Rightarrow Y$: c% of transactions with X also contain Y
- *Support* s of rule $X \Rightarrow Y$: s% of transactions in **D** contain $X \cup Y$
- TASK: Find all rules with minimum support *minsup* and minimum confidence *minconf*
  (not just verify that a *given rule* $X \Rightarrow Y$ realizes *minsup* and *minconf*!)

139

Algorithms for Association Rules Discovery

- Initial algorithm proposed by Agrawal, Imieliński, Swami (1993)
- Based on following facts:
  - $sup(X \cup Y) \leq sup(X)$
  - Support of rule $X \Rightarrow Y = sup(X \cup Y)$



Justification of fact 1:
$sup(X \cup Y) = |T_{X \cup Y}|/|D| = |T_X \cap T_Y|/|D| \leq |T_X|/|D| = sup(X)$

140

Association Rules Discovery

- Algorithm

Step 1: Find all candidate itemsets (i.e., itemsets with support $\geq$ *minsup,* denoted *large itemsets*)

Step 2: Using set of large itemsets generate all rules with *minconf*

Algorithm – Step 1 (Apriori)

<u>Find all *large itemsets*:</u>

1. Initialize set C_k of *candidate itemsets* of size k=1 to **I**
2. Prune C_k by eliminating itemsets with support < *minsup*
   - Scan set of transactions **D**
   - For each transaction increase counter for each element in C_k covered by that transaction
   - When done, eliminate elements of C_k with support < *minsup*
3. If $C_k = \emptyset$ then stop
4. Create C_{k+1} by extending C_k
5. k++; goto 2

Algorithm – Step 2

<u>Using large itemsets, generate rules with *minconf*</u>

For each large itemset L try all rules of the form:

$a \Rightarrow L - a$, where $a \subset L$

If $sup(L)/sup(a) \geq$ *minconf* then output rule $a \Rightarrow L - a$

Justification of this procedure:

- Rule will have minimum support because L is a large itemset
- $conf(X \Rightarrow Y) = |\mathbf{T}_{X \cup Y}| / |\mathbf{T}_X| = sup(X \cup Y)/sup(X)$
- Hence $conf(a \Rightarrow L - a) = sup(a \cup (L-a))/sup(a) = sup(L)/sup(a)$

143

Example – Transaction DB

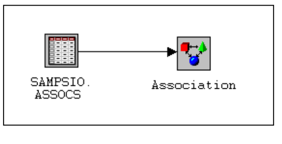
Layout of transaction DB

CUSTOMER – ID of a transaction

PRODUCT – ID of an item

(TIME not used)

Enterprise Miner diagram



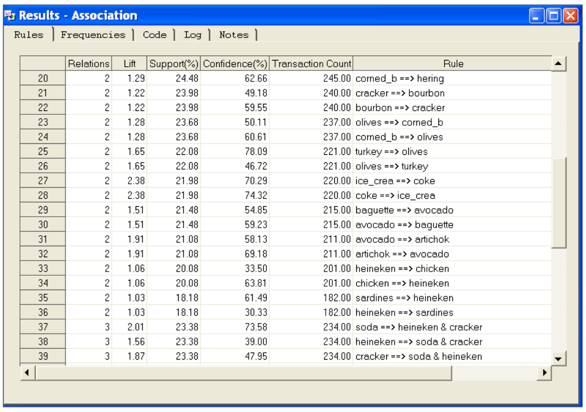
VIEWTABLE: Sampsio.Assocs

	CUSTOMER	TIME	PRODUCT
1	0	0	hering
2	0	1	corned_b
3	0	2	olives
4	0	3	ham
5	0	4	turkey
6	0	5	bourbon
7	0	6	ice_crea
8	1	0	baguette
9	1	1	soda
10	1	2	hering
11	1	3	cracker
12	1	4	heineken
13	1	5	olives
14	1	6	corned_b
15	2	0	avocado
16	2	1	cracker
17	2	2	artichok
18	2	3	heineken
19	2	4	ham
20	2	5	turkey
21	2	6	sardines
22	3	0	olives
23	3	1	bourbon
24	3	2	coke

144

Example – Rules Discovered

Parameters of discovered rules:
- Support
- Confidence
- Lift

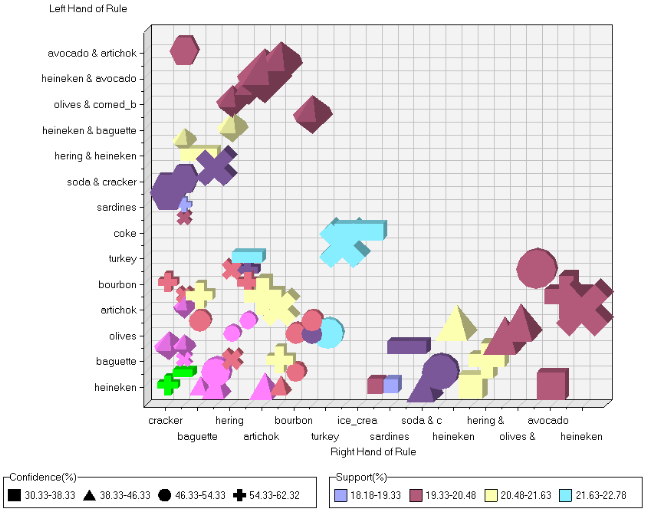


Results - Association

Rules | Frequencies | Code | Log | Notes

	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
20	2	1.29	24.48	62.66	245.00	corned_b ==> hering
21	2	1.22	23.98	49.18	240.00	cracker ==> bourbon
22	2	1.22	23.98	59.55	240.00	bourbon ==> cracker
23	2	1.28	23.68	50.11	237.00	olives ==> corned_b
24	2	1.28	23.68	60.61	237.00	corned_b ==> olives
25	2	1.65	22.08	78.09	221.00	turkey ==> olives
26	2	1.65	22.08	46.72	221.00	olives ==> turkey
27	2	2.38	21.98	70.29	220.00	ice_crea ==> coke
28	2	2.38	21.98	74.32	220.00	coke ==> ice_crea
29	2	1.51	21.48	54.85	215.00	baguette ==> avocado
30	2	1.51	21.48	59.23	215.00	avocado ==> baguette
31	2	1.91	21.08	58.13	211.00	avocado ==> artichok
32	2	1.91	21.08	69.18	211.00	artichok ==> avocado
33	2	1.06	20.08	33.50	201.00	heineken ==> chicken
34	2	1.06	20.08	63.81	201.00	chicken ==> heineken
35	2	1.03	18.18	61.49	182.00	sardines ==> heineken
36	2	1.03	18.18	30.33	182.00	heineken ==> sardines
37	3	2.01	23.38	73.58	234.00	soda ==> heineken & cracker
38	3	1.56	23.38	39.00	234.00	heineken ==> soda & cracker
39	3	1.87	23.38	47.95	234.00	cracker ==> soda & heineken

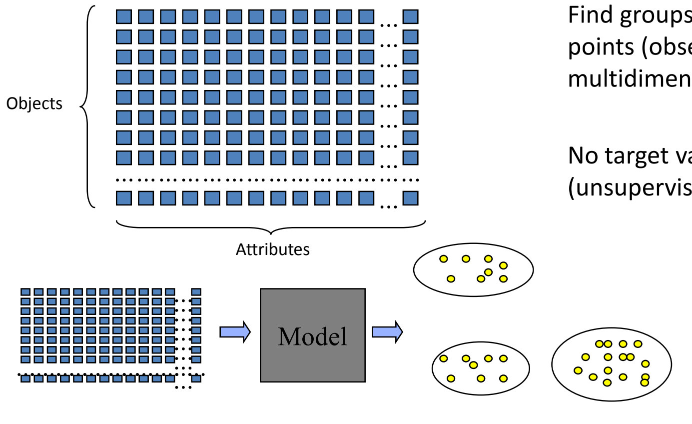
Example – Rules Discovered



Clustering Algorithms – Contents

- K-means
- Hierarchical algorithms
- Linkage functions
- Vector quantization

Clustering – Formulation



Find groups of *similar* points (observations) in multidimensional space

No target variable (unsupervised learning)

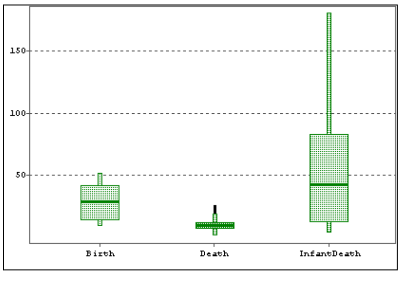
Methods of Clustering - Overview

- Variety of methods:
  - Hierarchical clustering – create hierarchy of clusters (one cluster entirely contained within another cluster)
  - Non-hierarchical methods – create disjoint clusters
  - Overlapping clusters (objects can belong to >1 cluster simultaneously)
  - Fuzzy clusters (defined by the probability (grade) of membership of each object in each cluster)
- Useful data preprocessing prior to clustering:
  - PCA (Principal components analysis) – to reduce dimensionality of data
  - Data standarization (transform data to reduce large influence of variables with larger variance on results of clustering)

149

Introductory Example

- 97 countries described by 3 attributes: Birth, Death, InfantDeath rate (given as number per 1000, data from year 1995)

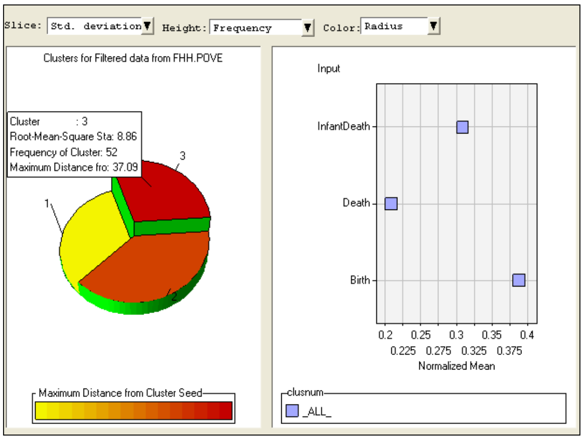


Birth	Death	InfantDeath	Country
24.7	5.7	30.8	Albania
12.5	11.9	14.4	Bulgaria
13.4	11.7	11.3	Czechoslovakia
12	12.4	7.6	Former_E._Germany
11.6	13.4	14.8	Hungary
14.3	10.2	16	Poland
13.6	10.7	26.9	Romania
14	9	20.2	Yugoslavia
17.7	10	23	USSR
15.2	9.5	13.1	Byelorussia_SSR
13.4	11.6	13	Ukrainian_SSR
20.7	8.4	25.7	Argentina
46.6	18	111	Bolivia
28.6	7.9	63	Brazil
23.4	5.8	17.1	Chile
27.4	6.1	40	Columbia
32.9	7.4	63	Ecuador
28.3	7.3	56	Guyana
34.8	6.6	42	Paraguay
32.9	8.3	109.9	Peru
18	9.6	21.9	Uruguay
27.5	4.4	23.3	Venezuela
29	23.2	43	Mexico
12	10.6	7.9	Belgium
13.2	10.1	5.8	Finland
12.4	11.9	7.5	Denmark
13.6	9.4	7.4	France

150

Example – cntd.

Analysis I
- Clustering raw data
- k-means algorithm
- Result: 3 clusters
(no. of obs. in each cluster: 13, 32, 52)

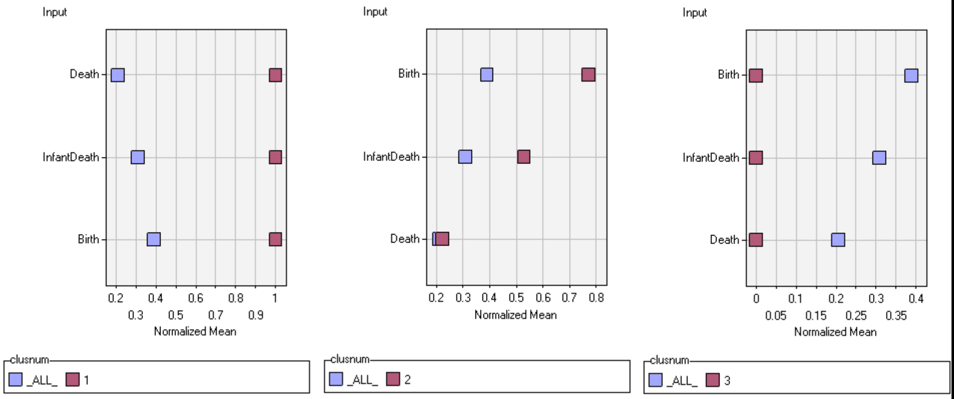


Birth	Death	InfantDeath	Country	Cluster ID
40.4	18.7	181.6	Afghanistan	1
42.2	15.5	119	Bangladesh	1
41.4	16.6	130	Cambodia	1
39.6	14.8	128	Nepal	1
47.2	20.2	137	Angola	1
48.6	20.7	137	Ethiopia	1
47.4	21.4	143	Gambia	1
48.3	25	130	Malawi	1
45	18.5	141	Mozambique	1
44	12.1	135	Namibia	1
48.2	23.4	154	Sierra_Leone	1
50.1	20.2	132	Somalia	1
46.8	12.5	118	Swaziland	1

Birth	Death	InfantDeath	Country	Cluster ID
46.6	18	111	Bolivia	2
28.6	7.9	63	Brazil	2
32.9	7.4	63	Ecuador	2
28.3	7.3	56	Guyana	2
32.9	8.3	109.9	Peru	2
42.5	11.5	108.1	Iran	2
42.6	7.8	69	Iraq	2
42.1	7.6	71	Saudi_Arabia	2
29.2	8.4	76	Turkey	2
30.5	10.2	91	India	2
28.6	9.4	75	Indonesia	2
36.1	8.8	68	Mongolia	2
30.3	8.1	107.7	Pakistan	2
31.8	9.5	64	Vietnam	2
35.5	8.3	74	Algeria	2
48.5	11.6	67	Botswana	2

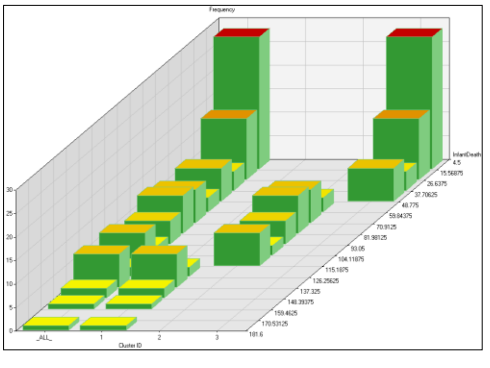
Birth	Death	InfantDeath	Country	Cluster ID
24.7	5.7	30.8	Albania	3
12.5	11.9	14.4	Bulgaria	3
13.4	11.7	11.3	Czechoslovakia	3
12	12.4	7.6	Former_E._Germany	3
11.6	13.4	14.8	Hungary	3
14.3	10.2	16	Poland	3
13.6	10.7	26.9	Romania	3
14	9	20.2	Yugoslavia	3
17.7	10	23	USSR	3
15.2	9.5	13.1	Byelorussia_SSR	3
13.4	11.6	13	Ukrainian_SSR	3
20.7	8.4	25.7	Argentina	3
23.4	5.8	17.1	Chile	3
27.4	6.1	40	Columbia	3
34.8	6.6	42	Paraguay	3
18	9.6	21.9	Uruguay	3
27.5	4.4	23.3	Venezuela	3
29	23.2	43	Mexico	3
12	10.6	7.9	Belgium	3
13.2	10.1	5.8	Finland	3

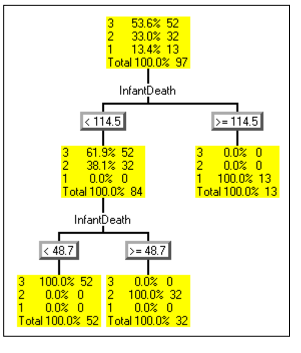
Example – Profiles of Clusters



Example – Profiles of Clusters

- Notice: data clustered based on InfantDeath Rate only!



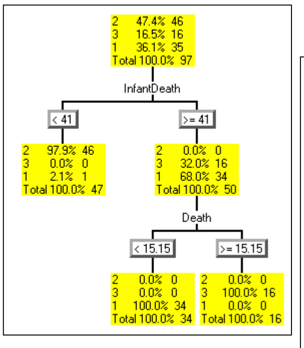


Example – Standarization of Data

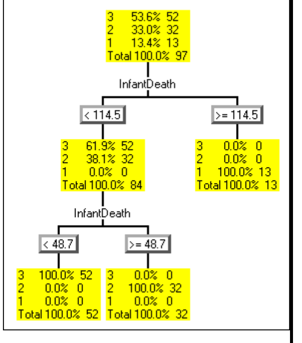
Analysis II

- Data standarized prior to clustering (variables divided by their standard deviation)
- Result: 3 clusters (with 35, 46, 16 obs.)
- Data clustered based on InfantDeath and Death
- Observe that data with largest variance have largest influence on results of clustering

Analysis II

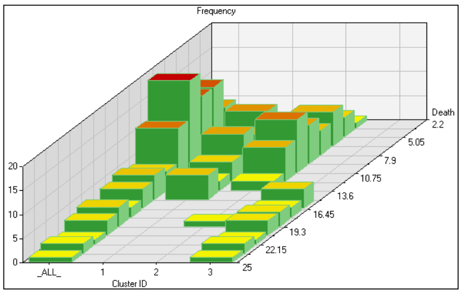


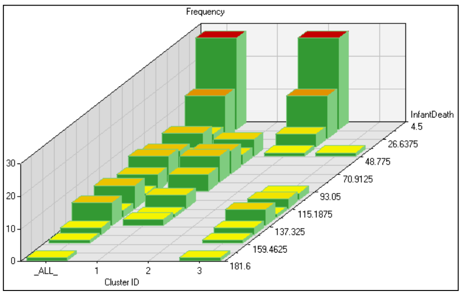
Analysis I



Example – Profiles of Clusters

- Analysis II: profiles of clusters





Methods of Clustering

- Non-hierarchical methods
  - K-means clustering
  - Non-deterministic
  - O(n) n - number of observations

- Hierarchical methods
  - Aglomerative (join small clusters)
  - Divisive (split big clusters)
  - Deterministic methods
  - $O(n^2) – O(n^3)$, depending on the clustering method (i.e. definition of inter-cluster distance)

157

Methods of Clustering - Remarks

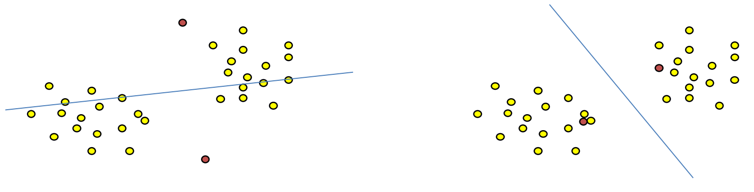
- Clustering large datasets
  - K-means
  - If results of hierarchical clustering needed – first use K-means yielding e.g. 50 clusters, followed by hierarchical clustering on results of K-means

- *Consensus clustering*
  - Discover *real* clusters in data – analyze stability of results with noise injected in data

158

K-means Algorithm

- K-means clustering
  - Select k points (centroids of initial clusters; select randomly)
  - Assign each observation to the nearest centroid (nearest cluster)
  - For each cluster find the new centroid
  - Repeat step 2 and 3 until no change occurs in cluster assignments



159

K-means Algorithm

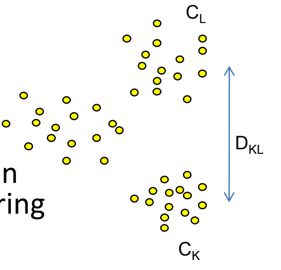
- Result: k separate clusters

- Algorithm requires that the correct number of clusters k is specified in advance
  (difficult problem: how to know the *real* number of clusters in data…)

160

Hierarchical Clustering

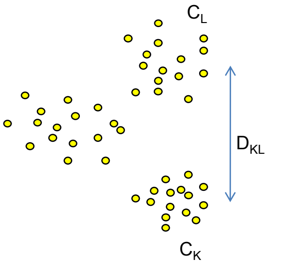
- Notation
  - x_i – observations, i=1..n
  - C_k – clusters
  - G – current number of clusters
  - D_{KL} – distance between clusters C_K and C_L
- Between-cluster distance D_{KL} – linkage function (various definitions available, results of clustering depend on D_{KL})



161

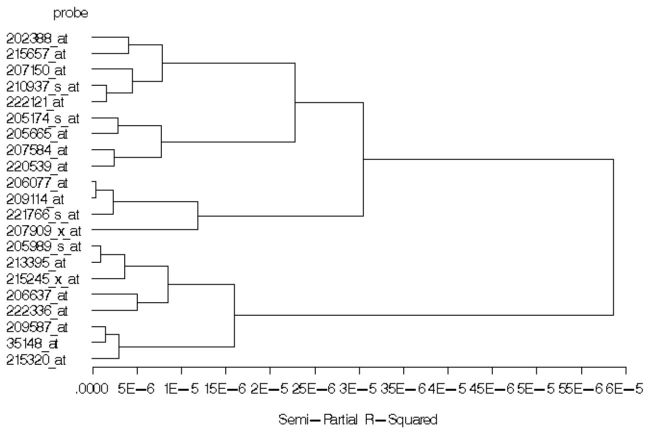
Hierarchical Clustering

- Algorithm (agglomerative hierarchical clustering)
  - $C_k = \{x_k\}$, k=1..n, G=n
  - Find K, L such that $D_{KL} = \min D_{IJ}$, $1 <= I,J <= G$
  - Replace clusters C_K and C_L by cluster $C_K \cup C_L$, G=G-1
  - Repeat steps 2 and 3 while G>1
- Result: hierarchy of clusters → dendrogram



162

Hierarchy of Clusters - Dendrogram



163

Definitions of Distance Between Clusters

- Different definitions of distance between clusters
  - Average linkage
  - Single linkage
  - Complete linkage
  - Density linkage
  - Ward’s minimum variance method
  - …

(SAS CLUSTER procedure accepts 11 different definitions of inter-cluster distance)

164

Average Linkage

- Notation
  - x_i – observations, i=1..n
  - d(x,y) – distance between observations (Euclidean distance assumed from now on)
  - C_k – clusters
  - N_K – number of observations in cluster C_K
  - D_{KL} – distance between clusters C_K and C_L
  - $mean_{CK}$ – mean observation in cluster C_K
  - $W_K = \sum |x_i - mean_{CK}|^2 \quad x_i \in C_K$ – variance in cluster

$$D_{KL} = \frac{1}{N_K N_L} \sum_{i \in C_K} \sum_{j \in C_L} d(x_i, x_j)$$

- Average linkage
  - Tends to join clusters with small variance
  - Resulting clusters tend to have similar variance

$$D_{KL} = \|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_L\|^2 + \frac{W_K}{N_K} + \frac{W_L}{N_L}$$

Complete Linkage

- Notation
  - x_i – observations, i=1..n
  - d(x,y) – distance between observations
  - C_k – clusters
  - N_K – number of observations in cluster C_K
  - D_{KL} – distance between clusters C_K and C_L
  - $mean_{CK}$ – mean observation in cluster C_K
  - $W_K = \sum |x_i - mean_{CK}|^2 \quad x_i \in C_K$ – variance in cluster

$$D_{KL} = \max_{i \in C_K} \max_{j \in C_L} d(x_i, x_j)$$

- Complete linkage
  - Resulting clusters tend to have similar diameter

Single Linkage

- Notation
  - x_i – observations, i=1..n
  - d(x,y) – distance between observations
  - C_k – clusters
  - N_K – number of observations in cluster C_K
  - D_{KL} – distance between clusters C_K and C_L
  - $mean_{CK}$ – mean observation in cluster C_K
  - $W_K = \sum |x_i - mean_{CK}|^2 \quad x_i \in C_K$ – variance in cluster

$$D_{KL} = \min_{i \in C_K} \min_{j \in C_L} d(x_i, x_j)$$

- Single linkage
  - Tends to produce elongated clusters, irregular in shape

167

Ward's Minimum Variance Method

- Notation
  - x_i – observations, i=1..n
  - d(x,y) – distance between observations
  - C_k – clusters
  - N_K – number of observations in cluster C_K
  - D_{KL} – distance between clusters C_K and C_L
  - $mean_{CK}$ – mean observation in cluster C_K
  - $W_K = \sum |x_i - mean_{CK}|^2 \quad x_i \in C_K$ – variance in cluster
  - $B_{KL} = W_M - W_K - W_L$ where $C_M = C_K \cup C_L$

$$D_{KL} = B_{KL} = \frac{\|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_L\|^2}{\frac{1}{N_K} + \frac{1}{N_L}}$$

- Ward's minimum variance method
  - Tends to join small clusters
  - Tends to produce clusters with similar number of observations

168

Density Linkage

- Notation
  - x_i – observations, i=1..n
  - d(x,y) – distance between observations
  - r – a fixed constant
  - f(x) – proportion of observations within sphere centered at x with radius r divided by the volume of the sphere (measure of density of points near observation x)

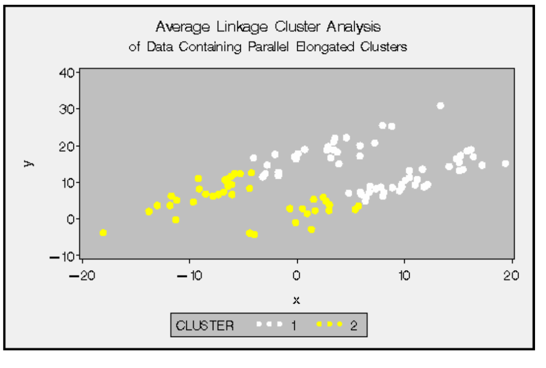
$$d^*(x_i, x_j) = \begin{cases} \frac{1}{2}\left(\frac{1}{f(x_i)} + \frac{1}{f(x_j)}\right) & \text{if } d(x_i, x_j) \le r \\ \infty & \text{otherwise} \end{cases}$$

- Density linkage
  - We realize single linkage using the measure d*
  - Capable of discovering clusters of irregular shape

169

Example – Average Linkage

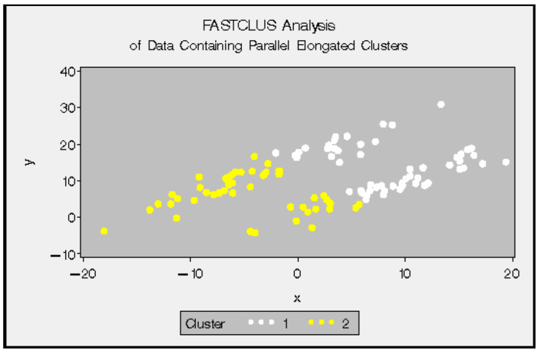
Elongated clusters in data



170

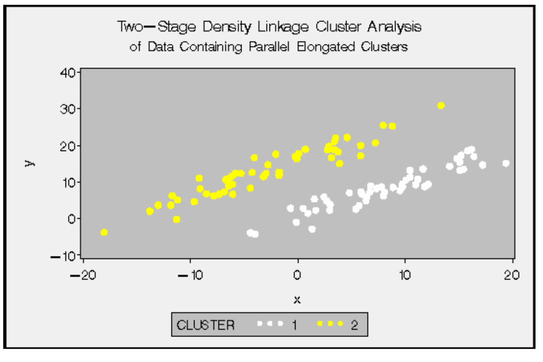
Example – K-means

Elongated clusters in data



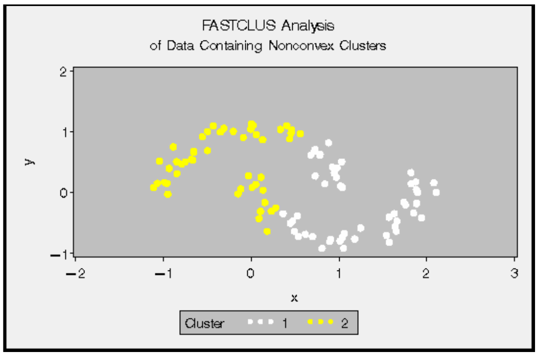
Example – Density Linkage

Elongated clusters in data



Example – K-means

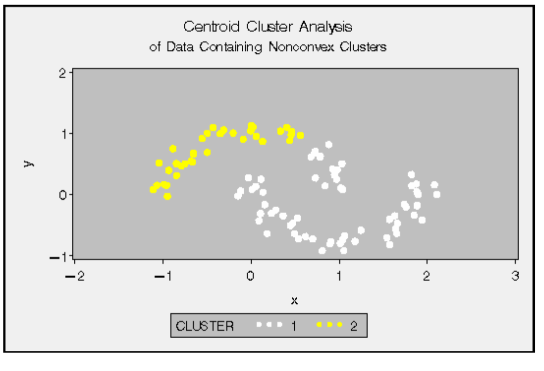
Nonconvex clusters in data



173

Example – Centroid Linkage

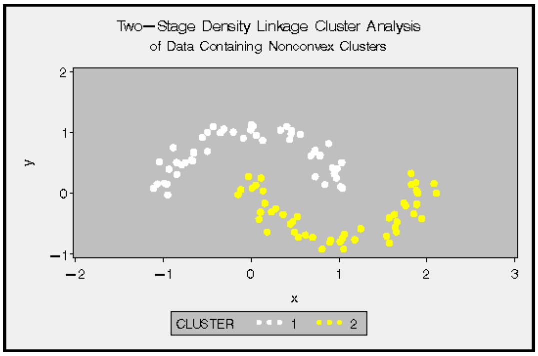
Nonconvex clusters in data



174

Example – Density Linkage

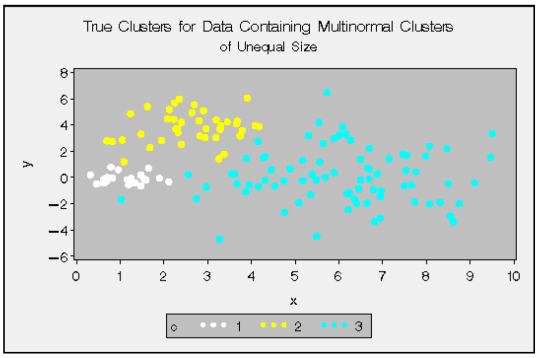
Nonconvex clusters in data



175

Example – True Clusters

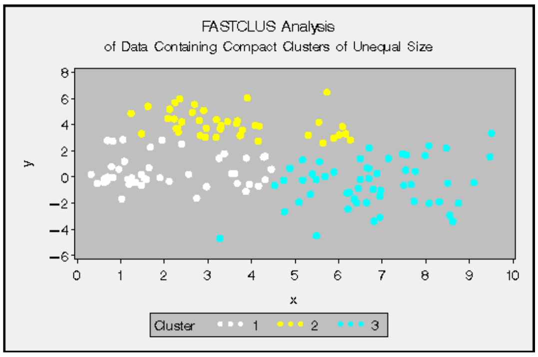
Clusters of unequal size



176

Example – K-means

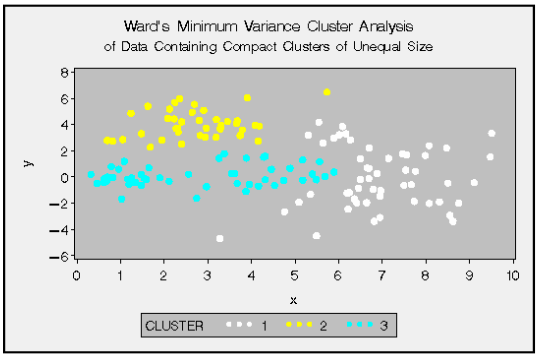
Clusters of unequal size



177

Example – Ward's Method

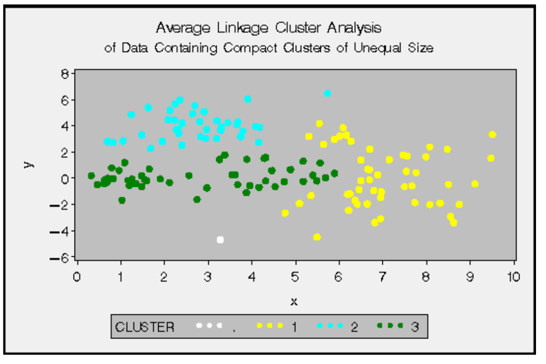
Clusters of unequal size



178

Example – Average Linkage

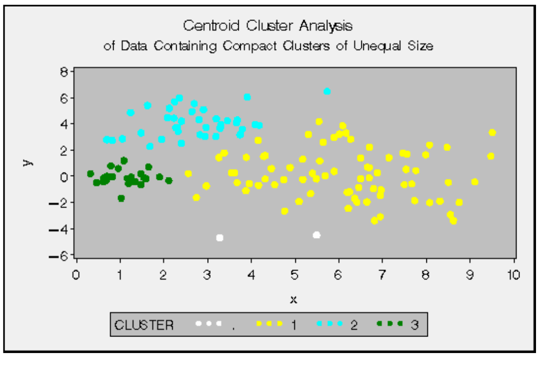
Method: average linkage



179

Example – Centroid Linkage

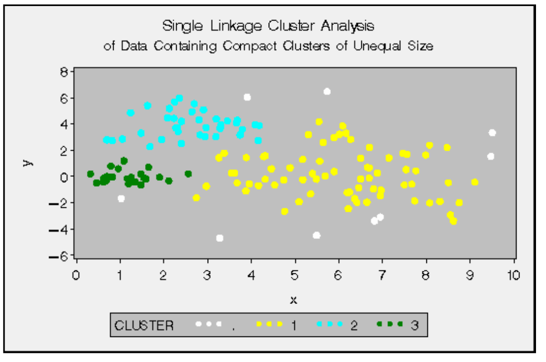
Clusters of unequal size



180

Example – Single Linkage

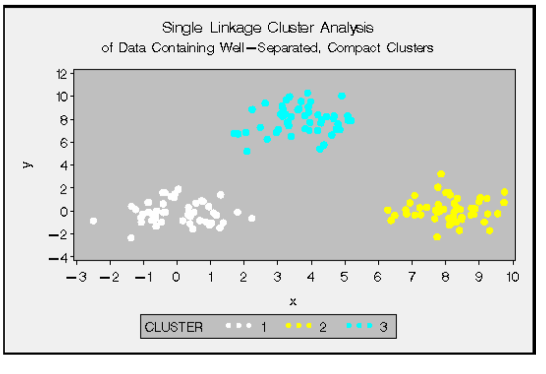
Clusters of unequal size



181

Example – Well Separated Data

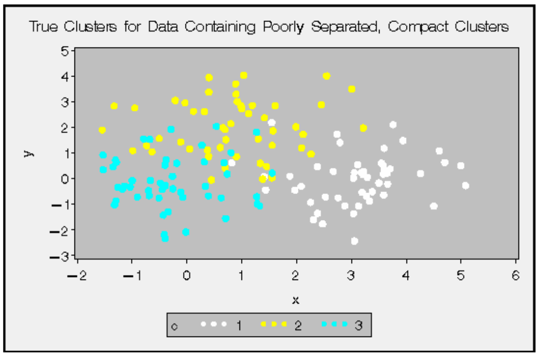
Any method will work ☺



182

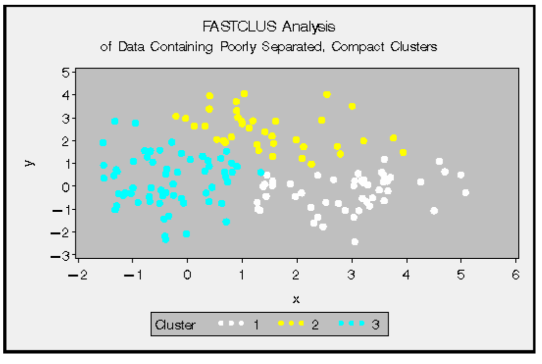
Example – Poorly Separated Data

True clusters



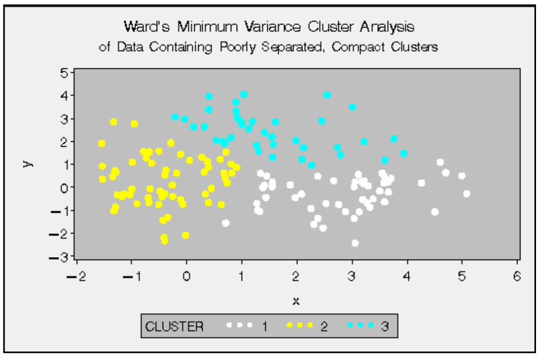
Example – Poorly Separated Data

Method: k-means



Example – Poorly Separated Data

Ward's method



185

Clustering Methods – Final Remarks

- Standarization of variables prior to clustering
  - Often necessary, otherwise variables with large variance tend to have large influence on clustering
  - Often standarized measurement z_{ij} is computed as the z-score:

$$z_{ij} = \frac{x_{ij} - \mu_j}{s_j}$$

    where x_{ij} – original measurement in observation *i* and variable *j*, μ_j – mean value of variable *j*, s_j – mean absolute deviation of variable *j* (or its standard deviation)

  - Other ideas: divide variable by its range, max value or standard deviation

186

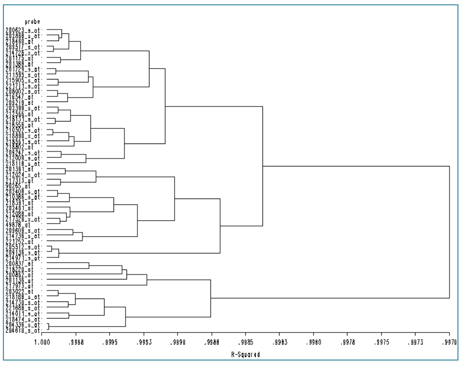
Clustering Methods – Final Remarks

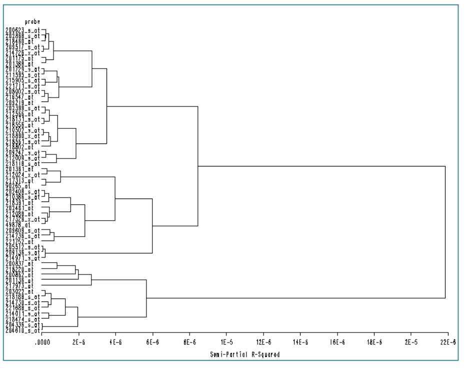
- The number of clusters
  - No satisfactory theory to determine the *right* number of clusters in data
  - Various criteria can be observed to help determine the *right* number of clusters, e.g. Criteria based on variance accounted for by clusters
    - $\mathbf{R^2}$=1-P_G/T
    - or **semipartial** $\mathbf{R^2}$=B_{KL}/T

    where T – total variance of observations; $P_G=\sum W_K$ over G clusters
    $B_{KL}=W_M-W_K-W_L$ where $C_M=C_K \cup C_L$
  - Cubic Clustering Criterion (CCC)
  - Often data visualization useful for determining the number of clusters
    - Scatterplot for 2-3 dimensional data
    - In high dimensions → apply PCA transformation (or similar) → visualize data in 2-3 dimensional space of first principal components

187

Example – R^2, Semi-partial R^2

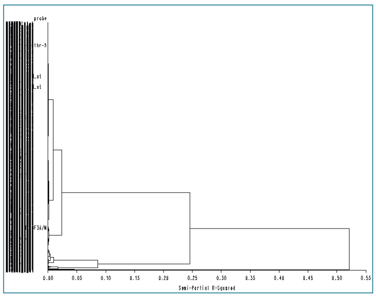




188

Example – Number of Clusters – Useful Checks

NCL	Clusters Joined		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2
20	CL37	CL35	795	0.0009	.980	.978	5.23	8917	500
19	CL52	CL47	186	0.0010	.979	.977	4.10	8958	177
18	CL25	203290_at	48	0.0010	.978	.977	3.24	9046	20.2
17	CL73	CL30	49	0.0011	.977	.976	2.43	9151	31.0
16	CL20	CL79	944	0.0011	.976	.975	1.92	9312	407
15	CL33	201426_s_at	29	0.0012	.975	.974	1.41	9485	25.7
14	CL66	CL74	5	0.0013	.973	.973	1.04	9694	19.5
13	CL38	CL19	345	0.0017	.972	.971	0.40	9876	196
12	CL22	CL41	142	0.0026	.969	.970	-1.5	9861	153
11	CL15	CL26	44	0.0037	.965	.968	-4.8	9644	36.6
10	CL24	CL27	9	0.0038	.961	.966	-6.9	9623	11.2
9	CL12	CL18	190	0.0039	.958	.963	-8.1	9794	89.6
8	CL21	CL16	2830	0.0087	.949	.959	-14	9216	4743
7	CL9	CL17	239	0.0098	.939	.955	-13	8931	157
6	CL11	CL23	57	0.0172	.922	.947	-18	8213	92.5
5	CL8	CL13	3175	0.0240	.898	.935	-21	7652	4141
4	CL10	CL14	14	0.0454	.852	.914	-25	6704	70.7
3	CL7	CL6	296	0.0862	.766	.866	-28	5708	487
2	CL5	CL3	3471	0.2453	.521	.731	-35	3789	4713
1	CL2	CL4	3485	0.5210	.000	.000	0.00	.	3789



- PST2: 3 or 6 or 9 (one before peak in value)
- PSF: 9 (peak in value)
- CCC: 18 (CCC around 3)

189

Kohonen VQ (Vector Quantization)

- Algorithm similar to k-means
- Idea of VQ algorithm:
  1. Select k points (initial cluster *centroids*)
  2. For observation x_i find nearest centroid (*winning seed*) – denoted by S_n
  3. Modify S_n according to the formula:

     $S_n=S_n(1-L)+x_iL$,

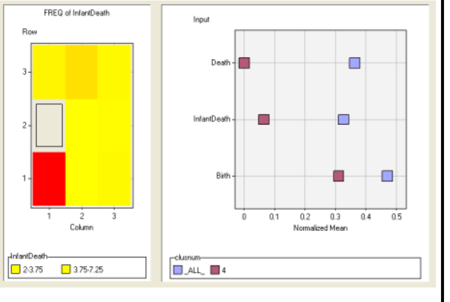
     where L – learning constant (decreased during learning)
  4. Repeat steps 2 and 3 over all training observations
  5. Repeat steps 2-4 given number of iterations

190

Kohonen SOM (Self Organizing Maps)

- Idea of the SOM algorithm

1. Select k initial points (cluster centroids), represent them on a 2D map
2. For observation x_i find winning seed S_n
3. Modify all centroids :
   S_j=S_j (1-K(j,n)L)+x_iK(j,n)L,
   where
   L – learning constant (decreasing during training)
   K(j,n) – function decreasing with increasing distance on the 2D map between S_j i S_n centroids (K(j,j)=1)
4. Repeat steps 2 and 3 over all training observations



191

Data Mining Tools

- SAS Enterprise Miner, SAS Foundation
  http://www.sas.com
- IBM SPSS Modeler, IBM SPSS Statistics
  http://www.spss.com , http://www-01.ibm.com/software/analytics/spss/
- Insightful Miner, RapidMiner, MS Analysis Services, …
- R System (free version of S-System)
  http://www.r-project.org
  http://www.cran.r-project.org (The Comprehensive R Archive Network
  http://www.bioconductor.org (Bioconductor)
- WEKA (free Data Mining software in Java)
  http://www.cs.waikato.ac.nz/ml/weka/
- scikit-learn (library for Python)
- knime (uses Weka)
  http://www.knime.org
- …

192