Data Mining

Marc M. VAN HULLE
K.U.Leuven
Faculteit Geneeskunde
Laboratorium voor Neuro- en Psychofysiologie
Campus Gasthuisberg, Herestraat
B-3000 Leuven, BELGIUM
E-mail: marc@neuro.kuleuven.be

Jesse DAVIS
K.U.Leuven
Department Informatics
Celestijnenlaan 200a - bus 2402
B-3001 Heverlee, BELGIUM
E-mail: jesse.davis@cs.kuleuven.be

1 Introduction

Overview:

1.1 Knowledge discovery in databases – data mining

Knowledge discovery in databases (KDD) $\triangleq$

non-trivial *process* of identifying *valid*, *novel*, *potentially useful & understandable patterns & relationships* in data

(knowledge = patterns & relationships)

- **pattern:** expression describing facts about data set
- **relation:** expression describing dependencies between data and/or patterns
- **process:** KDD is multistep process, involving data preparation, data cleaning, data mining. . . (see further)
- **valid:** discovered patterns, relationships should be valid on new data with some certainty (or correctness, below error level)
- **novel:** not yet known (to KDD system)
- **potentially useful:** should lead to potentially useful actions (lower costs, increased profit,. . .)
- **understandable:** provide knowledge that is understandable to humans, or that leads to a better understanding of the data set

1.1 KDD – data mining – Cont'd

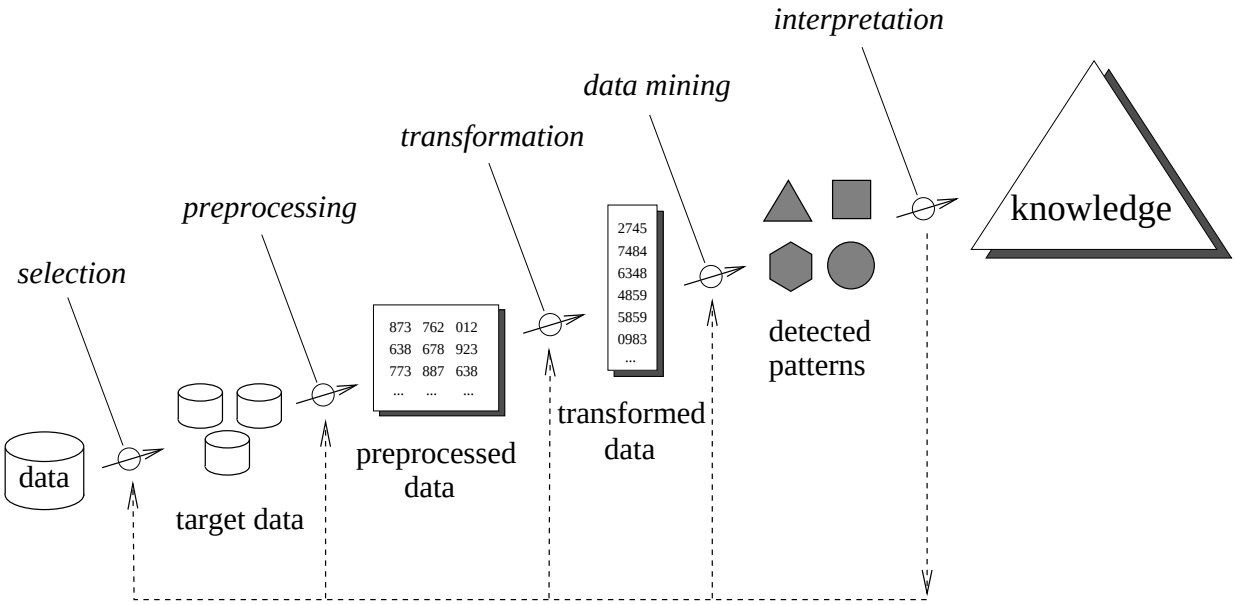
Data mining $\triangleq$

step in KDD process aimed at discovering patterns & relationships in preprocessed & transformed data

Hence: *knowledge discovery* =
data preparation + *data mining* + evaluation/interpretation of discovered patterns/relationships

Note: nowadays, data mining $\equiv$ KDD
data preparation,. . . = part of data mining *tool box*

1.2 Steps in knowledge discovery



Steps in KDD process

1. develop understanding of application domain, relevant prior knowledge, goals of end-user

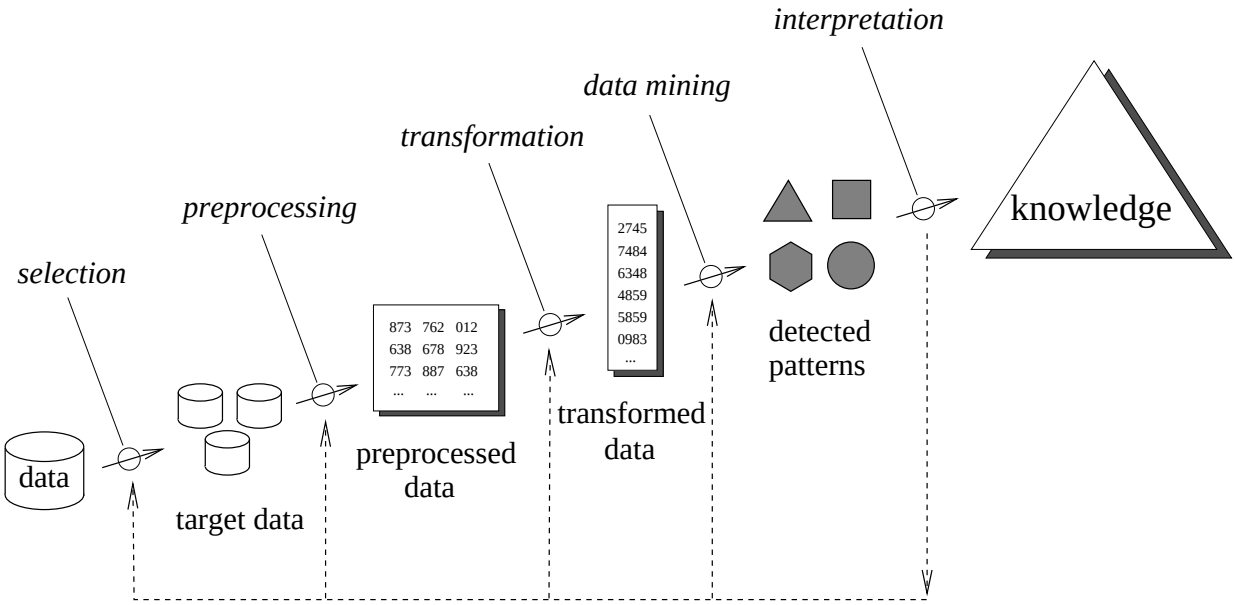
2. create target data set (= subset)

3. data cleaning and preprocessing: remove noise & outliers/wildshots,
   handle missing data & unlabeled data,. . .

4. transform data (dimensionality reduction & data projection):
   find useful features with which to more efficiently represent data

5. select data mining *task*

1.2 Steps in knowledge discovery – Cont'd



Steps in KDD process

6. choose data mining *algorithm*

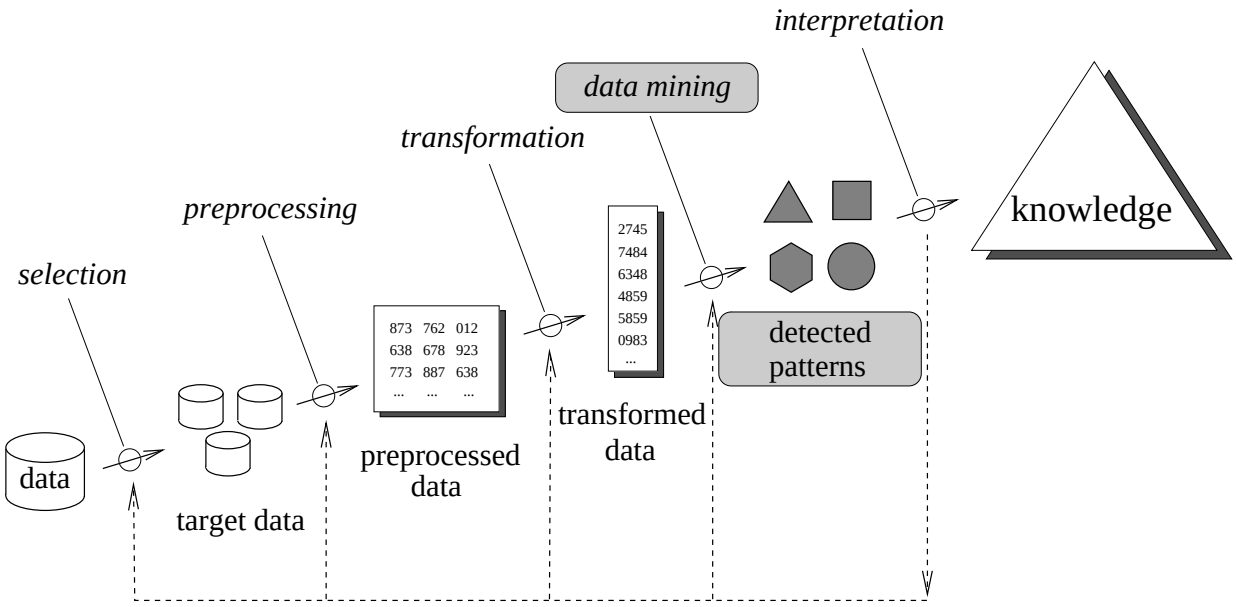
7. data mining

8. interpret the mined patterns, relationships
   → possible return to steps 1–7

9. consolidate discovered knowledge

1.3 Data Mining

Core process in KDD = data mining

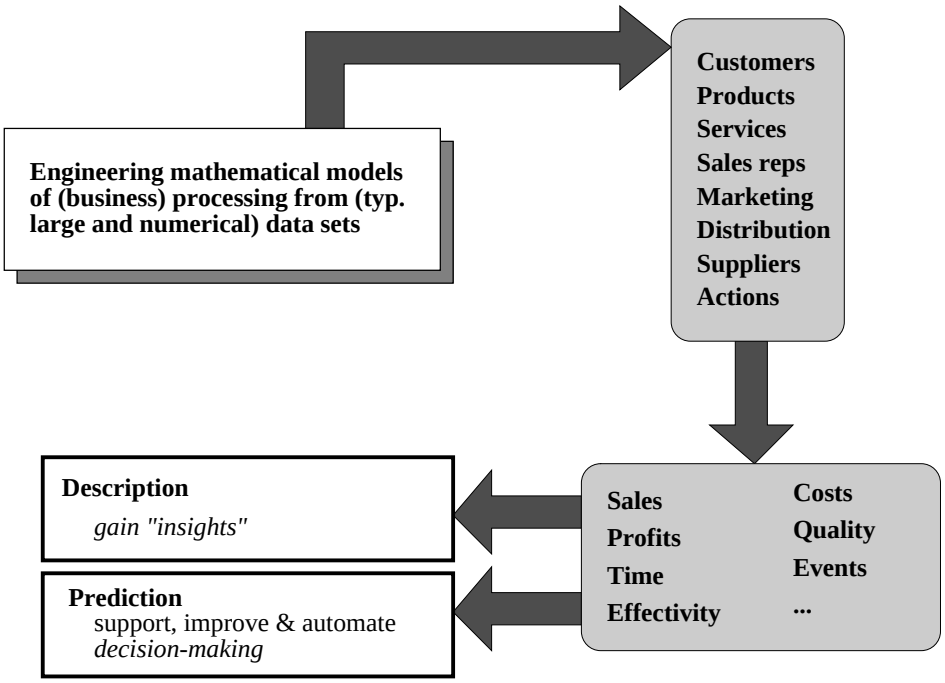


Steps in KDD process

1.3.1 Data Mining Objectives

Two "high-level" objectives of Data mining:
prediction & description

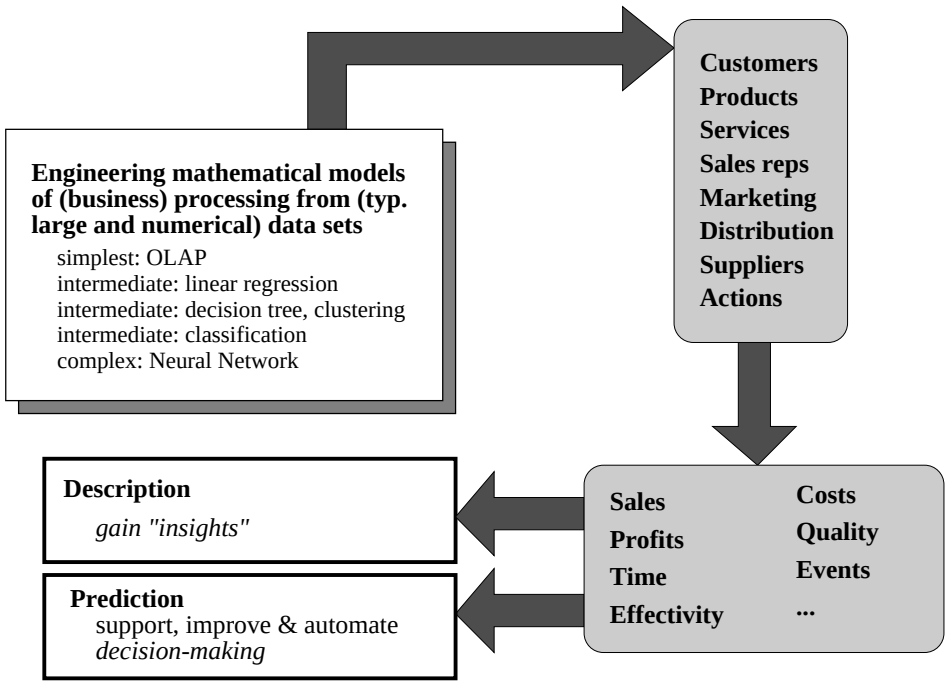
- *Prediction* of unknown or future values of selected variables
- *Description* in terms of (human-interpretable) patterns



Data mining objectives

1.3.1 Data Mining Objectives – Cont'd

- *Prediction & Description* involve modeling data set
- Differing degrees of model complexity:
  - simplest: OLAP
  - intermediate: linear regression, decision tree, clustering, classification
  - neural networks



OLAP = On-Line Analytical Processing

1.3.2 Building blocks of Data Mining

1.3.2.1 OLAP

- *On-Line Analytical Processing* (OLAP) $\stackrel{\triangle}{=}$ set of tools for providing multi-dimensional analysis of data warehouses

- *Data warehouse* $\stackrel{\triangle}{=}$ database that contains subject-oriented, integrated, and historical data, primarily used in analysis and decision support environments
  $\hookrightarrow$ requires collecting & cleaning transactional data & making it available for on-line retrieval
  = formidable task, especially in mergers with $\neq$ database archs.!

- OLAP = superior to SQL in computing summaries and breakdowns along dimensions
  (SQL (Standard Query Language) = script language for interrogating (manually) large databases such as `Oracle`)

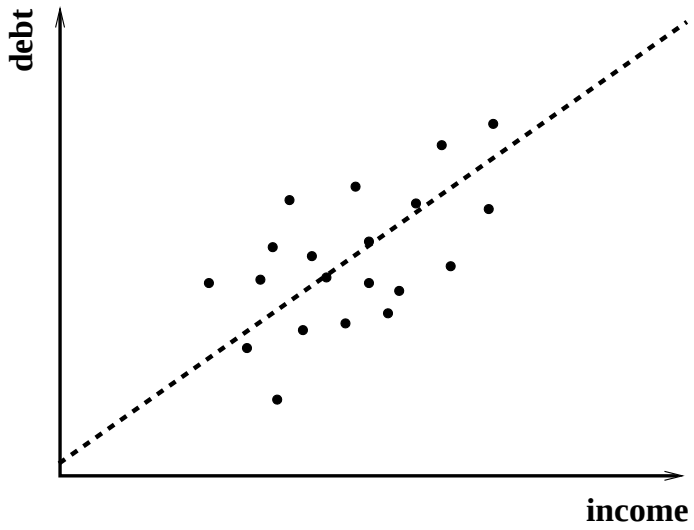
- OLAP requires substantial interaction from users to identify interesting patterns (clusters, trends)

- also: OLAP often confirms user's hunch
  $\neq$ looking for real "hidden" patterns, relations

- OLAP is now integrated into more advanced data mining tools

1.3.2.2 Linear regression

- Consider income/debt data

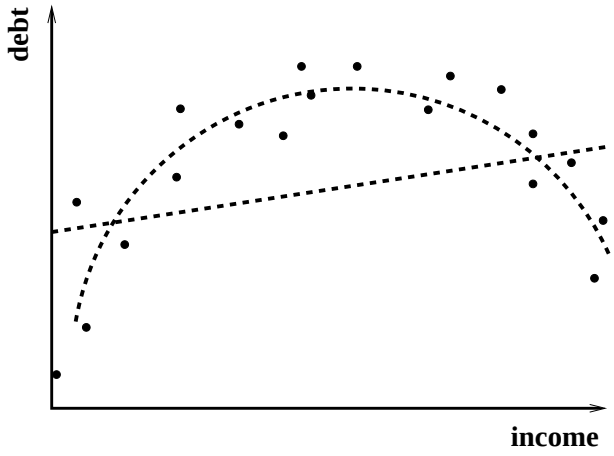


Linear regression example

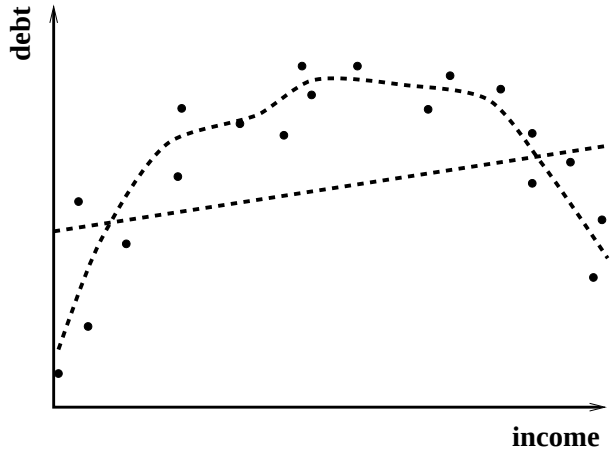
- **Assumptions:**
  1. independent variable = income
  2. dependent variable = debt
  3. relation between income & debt = **linear**

1.3.2.2 Linear regression – Cont'd

- **Failure:** when relation $\neq$ linear



Non-linear regression with semi-circle



Non-linear regression with smoothing spline

- **Hence:** need for non-linear regression capabilities

1.3.2.2 Linear regression – Cont'd

Type of regression models:

1. **functional** (descriptive) models: purpose is to summarize data compactly, not to explain system that generated data

2. **structural** (mechanistic) models: purpose is to account for physics, statistics,. . . of system that generated data

↪ best results obtained when **a priori knowledge** of system/process that generated data (structural modeling)

Example: estimate probability that drug X will cure patient
→ better model if imply knowledge of:
1) components of drug that curative/side effects
2) certainty of patient's diagnose (alternative diagnoses?)
3) patient's track record on reaction to drug components

1.3.2.2 Linear regression – Cont'd

What is the right level of model detail?

- more parameters $\neq$ better model!
  more parameters = more data needed to estimate them!

- more parameters = risk of overfitting!
  complex regression model goes through all data points,
  but fails to correctly model new data points !
  "With 5 parameters you can fit an elephant.
  And with 6 parameters you can make it blink!"

- more parameters $\neq$ deeper understanding of system/process that generated data
  cf. Occam's razor: simplest model that explains data = preferred
  leads to better understanding

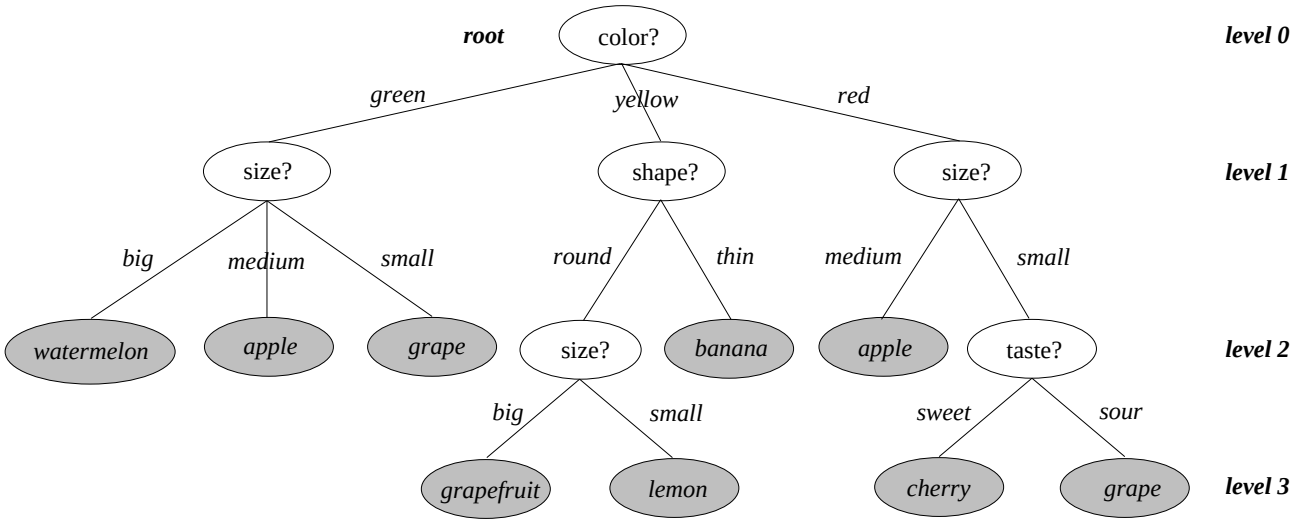
- good model has good predictive power
  (*e.g.*, by testing on new data points)

- good model provides confidence levels or degrees of certainty with regression results

1.3.2.3 Decision trees

- **Decision tree** $\stackrel{\triangle}{=}$ technique that recursively splits/divides space into $\neq$ (sub-)regions with decision hyperplanes orthogonal to coordinate axes

- Decision tree = root node + successive directional *links/branches* to other nodes, until *leaf node* reached
  - at each node: ask for **value** of particular **property** *e.g.*, color? green
  - continue until no further questions = leaf node
  - leaf node carries **class label**
  - test pattern gets class label of leaf node attached



Decision tree

1.3.2.3 Decision trees – Cont'd

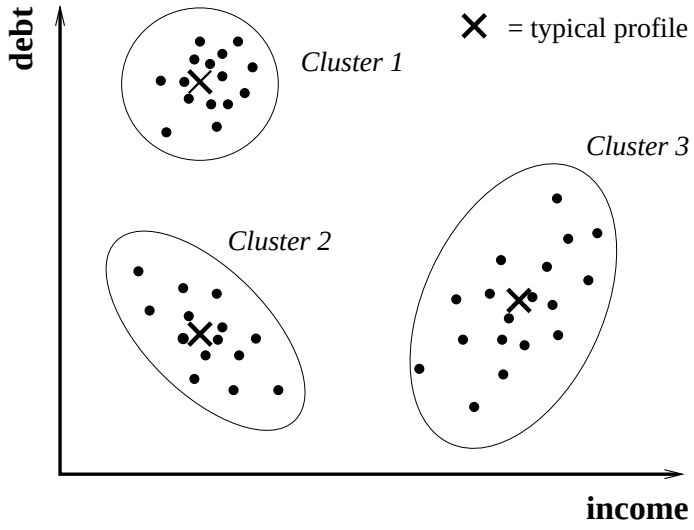
- **Advantages:**
  - easy to interpret
  - rules can be derived form tree:
    e.g., **Apple** = (`medium size` AND NOT `yellow color`)
- **Disadvantages:**
  - devours data at a rate exponential with depth
    hence: to uncover complex structure, extensive data needed
  - crude partitioning of space:
    corresponds to (hierarchical) classification problem in which each variable has different constant value for each class, independently from other variables

    (Note: classification = partitioning of set into subsets based on knowledge of class membership)

↪ Introduced by work on Classification And Regression Trees (CART) of Breiman *et al.* (1984)

1.3.2.4 Clustering

- **Clustering** = way to detect subsets of "similar" data
- **Example:** customer database:
  1. how many types of customers?
  2. what is typical customer profile for each type?
     e.g., cluster mean or median,. . . = cluster prototype



Clustering & cluster prototypes

1.3.2.4 Clustering – Cont'd

- wealth of clustering algorithms exist
  for overviews, see: Duda & Hart (1973), Duda *et al.* (2001), Theodoridis and Koutroumbas (1998)

- major types of clustering algorithms:

  1. **distortion-based clustering**
     = most widely used technique
     here: k-means & Fuzzy k-means clustering

  2. **density-based clustering**
     local peaks in density surface indicate cluster(-centers)
     here: hill-climbing

1.3.2.4 Clustering – Cont'd

1. Distortion-based clustering

k-means clustering

- **Assume:** we have c clusters & sample set $\mathcal{D} = \{\mathbf{v}_i\}$
- **Goal:** find mean vectors of clusters, $\mu_1, \ldots, \mu_c$
- **Algorithm:**

  1. initialize $\mu_1, \ldots, \mu_c$
  2. do for each $\mathbf{v}_j \in \mathcal{D}$, determine: $\arg\min_i \|\mathbf{v}_j - \mu_i\|$
     recompute $\forall i : \mu_i \leftarrow$ average{all samples $\in$ cluster$i$}
  3. until no change in μ_i, $\forall i$
  4. stop

- Converges with less iterations than number of samples
- Each sample belongs to exactly 1 cluster
- **Underlying idea:** minimize mean squared error (MSE) distortion
  (squared Euclidean distance) between cluster mean & cluster samples:

$$J_{k-means} = \sum_{i=1}^{c} \sum_{j=1}^{n} \mathcal{M}_i(\mathbf{v}_j) \|\mathbf{v}_j - \mu_i\|^2$$

  with $\mathcal{M}_i(\mathbf{v}_j) = 1$ if $i = \arg\min_k \|\mathbf{v}_j - \mu_k\|$, else $= 0$

1.3.2.4 Clustering – Cont'd

1. Distortion-based clustering – Cont'd

k-means clustering – Cont'd

- Cluster membership function $\triangleq$

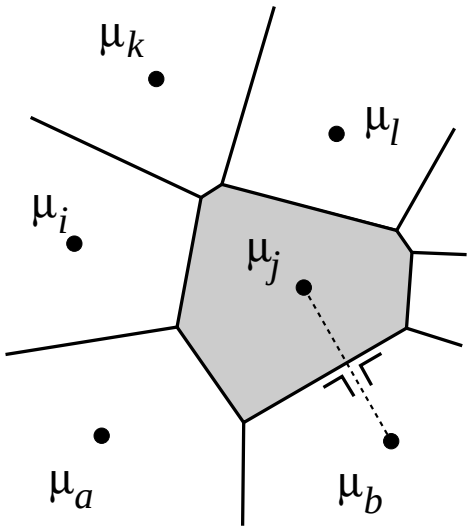
$$c = \arg \min_{\text{all clusters}} \|\mathbf{v}_j - \mu_i\|$$

  i.e., closest cluster prototype in Euclidean distance terms

- **Result:** partitioning of input space into non-overlapping regions *i.e.*, **quantization regions**

- Shape of quantization regions = **convex polytopes**, boundaries $\perp$ bisector planes of lines joining pairs of prototypes

- Partitioning $\triangleq$ **Voronoi tessellation** or **Dirichlet tessellation**

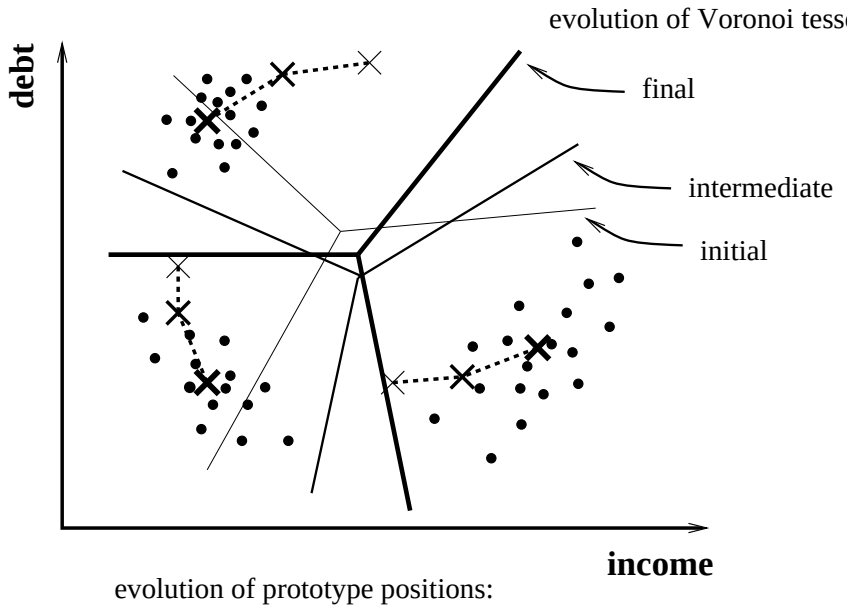


1.3.2.4 Clustering − Cont'd

1. Distortion-based clustering − Cont'd

k-means clustering − Cont'd

Example



evolution of prototype positions:

× = initial

× = intermediate

× = final

Trajectories of prototypes & Voronoi tessellation of the k-means clustering procedure on 3 clusters in 2D space

1.3.2.4 Clustering – Cont'd

1. Distortion-based clustering – Cont'd

Fuzzy k-means clustering

- **Assume:** we have c clusters & sample set $\mathcal{D} = \{\mathbf{v}_i\}$
- Each sample belongs in **probability** to cluster
  each sample has graded or "fuzzy" cluster membership
- **Define:** $P(\omega_i|\mathbf{v}_j, \theta)$ is probability that sample $\mathbf{v}_j$ belongs to cluster i, given θ parameter vector of membership functions, $\theta = \{\mu_1, \ldots, \mu_c\}$
  (we further omit θ in our notation)
- **Note that:** $\sum_{i=1}^{c} P(\omega_i|\mathbf{v}_j) = 1$, $\forall \mathbf{v}_j \in \mathcal{D}$ (*i.e.*, normalized)
- **Goal** is to minimize:

$$J_{fuzz} = \sum_{i=1}^{c} \sum_{j=1}^{n} [P(\omega_i|\mathbf{v}_j)]^b \|\mathbf{v}_j - \mu_i\|^2$$

  by gradient descent, $\frac{\partial J_{fuzz}}{\partial \mu_i}$

  Note:

  $b = 0$ *cf.* MSE minimization

  $b > 1$ each pattern belongs to several classes, *e.g.*, $b = 2$

1.3.2.4 Clustering – Cont'd

1. Distortion-based clustering – Cont'd

Fuzzy k-means clustering – Cont'd

- **Result of gradient descent:**
  μ_i is computed at each iteration step as:

$$\mu_i \leftarrow \frac{\sum_{j=1}^{n}[P(\omega_i|\mathbf{v}_j)]^b \mathbf{v}_j}{\sum_{j=1}^{n}[P(\omega_i|\mathbf{v}_j)]^b}, \quad \forall i$$

  $P(\omega_i|\mathbf{v}_j)$ is computed as:

$$P(\omega_i|\mathbf{v}_j) \leftarrow \frac{(\frac{1}{d_{ij}})^{\frac{1}{b-1}}}{\sum_{r=1}^{c}(\frac{1}{d_{rj}})^{\frac{1}{b-1}}}$$

  with $d_{ij} = \|\mathbf{v}_j - \mu_i\|^2$

- **Algorithm:**

  1. initialize $\mu_1, \ldots, \mu_c$, $P(\omega_i|\mathbf{v}_j)$, $\forall i, j$
  2. do recompute μ_i, $\forall i$
     recompute $P(\omega_i|\mathbf{v}_j)$, $\forall i, j$
  3. until small change in μ_i, $\forall i$, $P(\omega_i|\mathbf{v}_j)$, $\forall i, j$
  4. stop

1.3.2.4 Clustering – Cont'd

1. Distortion-based clustering – Cont'd

Advantages/disadvantages

- **advantage:** simple to implement + wealth of *heuristics*
- **disadvantages:**
  1. assumes cluster distribution = spherical around center
  2. assumes number of clusters = known

$\hookrightarrow$ heuristics developed for determining number of clusters

1.3.2.4 Clustering – Cont'd

1. Distortion-based clustering – Cont'd

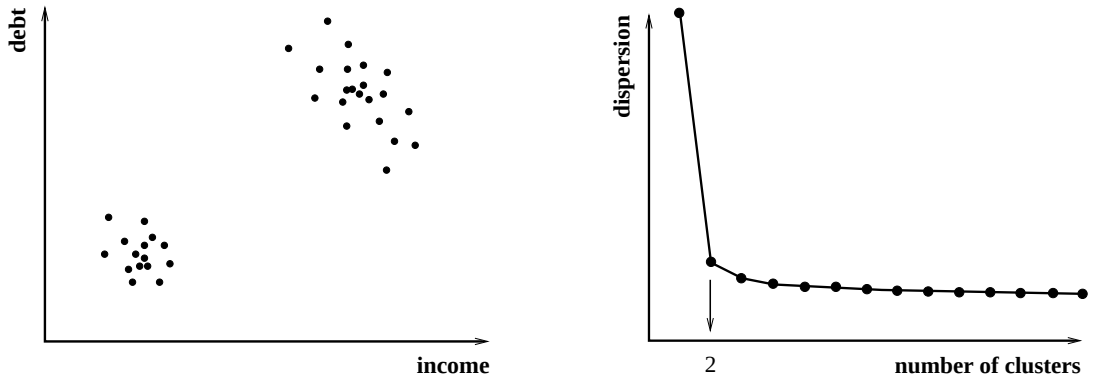
Heuristics for "optimal" number of clusters

1) "Statistical folklore technique":

1. *cluster dispersion* $\stackrel{\Delta}{=}$ sum of squared Euclidean distances between pairs of samples of same cluster:

$$W_c = \sum_{r=1}^{c} \frac{1}{2n_r} \sum_{i,j \in \text{cluster}r} \|\mathbf{v}_i - \mathbf{v}_j\|^2$$

   with n_r = number of samples in rth cluster
2. plot cluster dispersion as function of number of clusters
3. look for point where dispersion decreases at slower rate
4. choose this point as "optimal" number of clusters



Left panel: 2 clusters.
Right panel: cluster dispersion as a function of number of clusters

1.3.2.4 Clustering – Cont'd

1. Distortion-based clustering – Cont'd

Heuristics for "optimal" number of clusters – Cont'd

2) Gap statistic (Tibshirani *et al*, 2000):

1. determine gap statistic:

$$Gap(c) = E(\log(W_c^*)) - \log(W_c)$$

$E(\log(W_c^*))$ = dispersion expected for *uniform distribution* with same range as original data set
(in fact, $E(\log(W_c))$ is average for B uniform data sets)

2. plot $Gap(c)$ as a function of number of clusters c

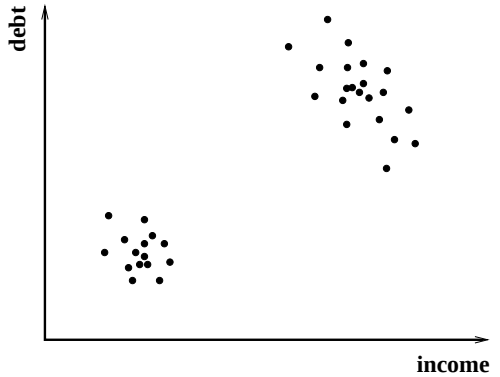
3. "optimal" number of clusters:

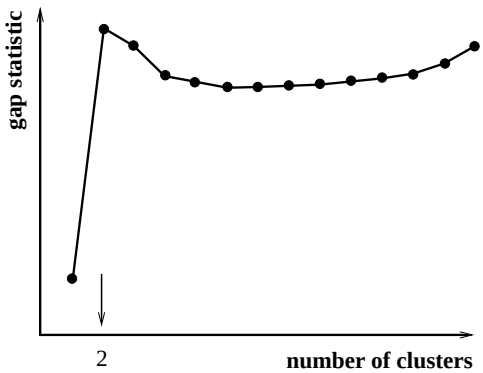
$$c_{\text{opt}} \triangleq c \text{ for which } Gap(c) \geq Gap(c+1) - s_{c+1}$$

with $s_{c+1} = sd_{c+1}\sqrt{1+\frac{1}{B}}$

with $sd_{c+1} = \left(\frac{1}{B}\sum_{b=1}^{B}\left(\log(W_c^{*b}) - \frac{1}{B}\sum_{b=1}^{B}\log(W_c^{*b})\right)^2\right)^{\frac{1}{2}}$

(*i.e.*, standard deviation of dispersions observed)





1.3.2.4 Clustering – Cont'd

2. Density-based clustering

- Determine **density estimate** at data points in set $\mathcal{D} = \{\mathbf{v}_i\}$
- **Hill-climbing:**
  1. since data estimate can be noisy, with irrelevant bumps:
     $\Rightarrow$ for each data point $\mathbf{v}_i$, look for higher density estimate within certain distance $\mathcal{K}$ from $\mathbf{v}_i$
  2. move to this new data point $\mathbf{v}_j$
     repeat operation until no further progress is possible:
     resulting data point = locus of **local density peak**
- Set of these loci = **cluster prototypes**
  set of data points $\rightsquigarrow$ same density peak = data points $\in$ **cluster**
- **Algorithm with hill-climbing:**

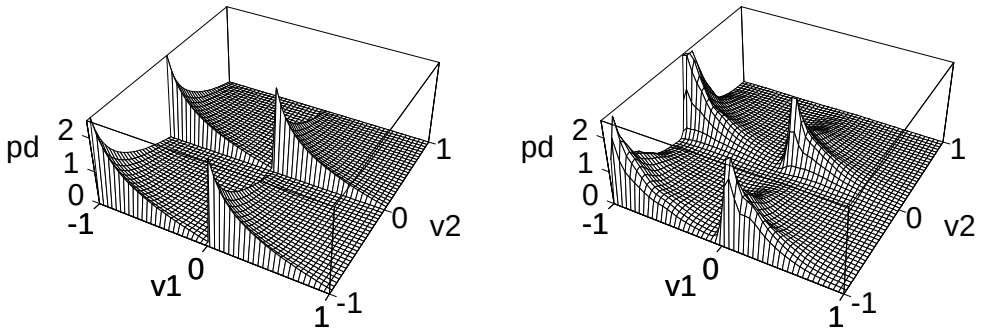
  1. determine density estimate given $\mathcal{D}$, $HC \leftarrow \mathcal{D}$, $HC2 \leftarrow \emptyset$
  2. do until HC does not change
  3. do $\forall$ data points $\in HC$
     look for data point $\mathbf{v}_j$ in range $\mathcal{K}$ with higher density
     $HC2 \leftarrow \mathbf{v}_j$
     $HC \leftarrow HC2$
     $HC2 \leftarrow \emptyset$
  4. stop

- **Note:** hill-climbing $\leftrightarrow$ gradient descent
- **Other algorithms:** valley-seeking approach (Koontz & Fukunaga, 1972), SKeleton by Influence Zones (SKIZ) (Serra, 1982),. . .

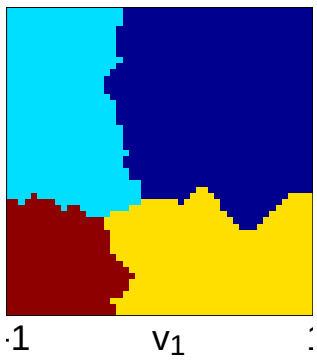
1.3.2.4 Clustering – Cont'd

2. Density-based clustering – Cont'd

Hill-climbing example



Left panel: Distribution from which samples are drawn
Right panel: Estimate of distribution determined from sample set



Clusters & boundaries determined with hill-climbing algorithm

1.3.2.4 Clustering – Cont'd

2. Density-based clustering – Cont'd

Advantages/disadvantages

- **advantage:** no assumptions about shape of density distribution
  $\rightsquigarrow$ less heuristics-based (more objective)

- **disadvantages:**

  1. vulnerable in high dim. spaces
     difficulty estimate density $\uparrow\uparrow$ when dimensionality $\uparrow$

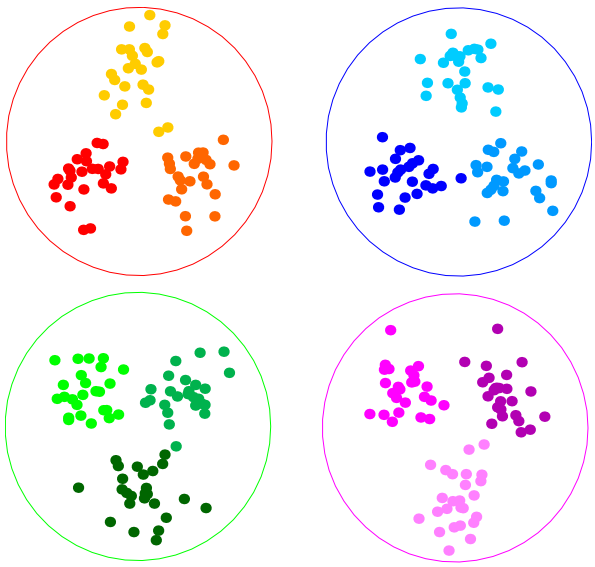
  2. how to determine optimal smoothness of density estimate?
     $\rightsquigarrow$ complex/tricky procedures

1.3.2.4 Clustering – Cont'd

Hierarchical clustering

Clusters arranged in tree

- divisive clustering (progressive subdivision of data set)
- agglomerative clustering (progressively merge clusters)

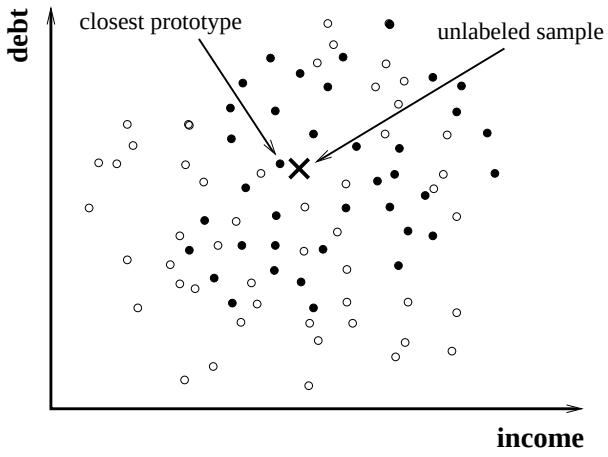


Divisive clustering

1.3.2.5 Classification

1) Nearest-Neighbor classification

- simplest classification technique
- **algorithm:**
  1. given set of samples (prototypes) & corresponding labels: $\mathcal{F} = \{(\mu_i, label_i)\}$
  2. what is label of sample $\mathbf{v}_j$?
  3. determine closest prototype:
$$c = \arg\min_i \|\mathbf{v}_j - \mu_i\|$$
  4. assign $label_c$ to sample $\mathbf{v}_j$



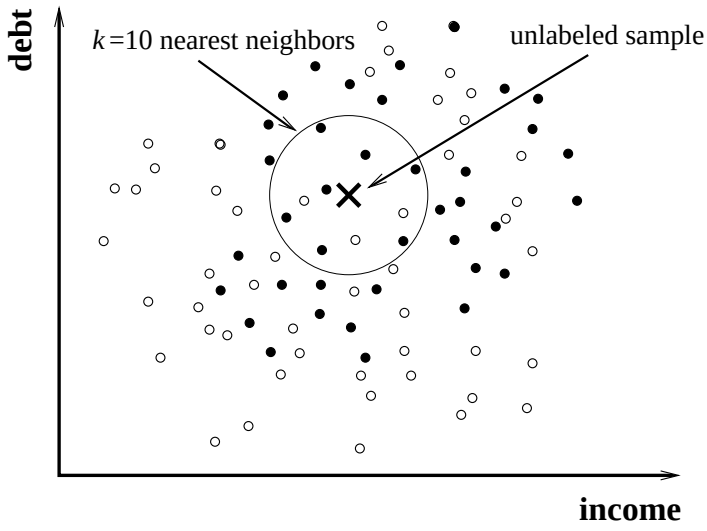
$\hookrightarrow$ assumes Voronoi tessellation of input space

- misclassifaction rate $>$ Bayesian rate (= min. misclass. rate) given ∞ # prototypes: misclass. $< 2 \times$ Bayesian rate

1.3.2.5 Classification – Cont'd

2) *k*-Nearest-Neighbor classification

- for $k > 1$: closer to Bayesian rate than $k = 1$ case
- **algorithm:**
  1. given set of samples (prototypes) & corresponding labels: $\mathcal{F} = \{(\mu_i, label_i)\}$
  2. what is label of sample $\mathbf{v}_j$?
  3. determine k closest prototypes
  4. label sample $\mathbf{v}_j$ according to majority of labels in k closest prototypes



$k = 10$ *nearest-neighbor classification:*
$7 \times \bullet$, $3 \times \circ$; *hence, sample gets* $\bullet$ *label*

1.3.2.6 Neural Networks

- **(Artificial) Neural Network** $\stackrel{\triangle}{=}$ network of nodes (**neurons**) that compute their states numerically through interactions via directional links (synaptic connections/**synapses**) with other nodes

- Synapses are modified so that network performs given task
  Modification is done with **learning algorithm**

- Network **architecture** + **learning algorithm**
  specify **type of task** network can solve

- in **Data Mining** ANNs are used for:
  - regression
  - classification
  - clustering
  - (time-series)prediction
  - feature extraction (PCA, ICA)
  - manifold projection (topographic maps, visual mining)

$\hookrightarrow$ for more information:
see courses Neural Computing (H02B3A) & Artificial Neural Networks (H02C4A)

1.3.3 Data Mining Topics of this Course

Data mining techniques:

- **Frequent pattern discovery**
  Find all patterns for which there are sufficient examples in the sample data.
  In contrast, k-optimal pattern discovery techniques find the k patterns that optimize a user-specified measure of interest. The value k is also specified by the user (*e.g.*, k-means clustering).

- **Graph mining**
  Structure mining or structured data mining is the process of finding and extracting useful information from semi-structured data sets. Graph mining is a special case of structured data mining.

- **Data stream mining**
  Paradigms for knowledge discovery from evolving data.

- **Visual mining**
  Exploratory data analysis by visualization of high-dim. data:
  - Data transformation techniques (PCA, ICA, MDS, SOM, GTM)
  - Iconic display techniques (abstract & metaphoric glyphs)

Data preparation techniques:

- Data transformation (dimensionality reduction)
- Feature extraction & -selection

1.3.4 Data Mining Tool Classification

Data mining tool vendors offer $\neq$ products:

- extended business intelligence (BI) suites
- generic data mining tool suites
- specialized stand-alone products
- proprietary solutions

$\hookrightarrow$ Data Mining tool classification

1.3.4 DM Tool Classification – Cont'd

1) Generic, application-independent tools

- wide array of data mining & visualization techniques: from clustering & regression to neural networks
- requires skilled staffing: have to know how to prepare data, what technique to use for given task, how to use technique, how to validate results, how to apply in business
- examples: IBM's *Intelligent Miner*, SPSS' *Clementine*, SAS Institute's *Enterprise Miner*

2) Algorithm-specific tools

- similar to *Generic tools*, but use specific set of algorithms
- yield better results if algorithms suited for given problem
- examples focussing on **decision trees**: *CART* from Salford Systems, *KnowledgeSeeker* from Angoss Software, *Alice* from Isoft
- examples focussing on **neural networks**: *NeuralWorks Profesional* from NeuralWare, *Viscovery* by Eudaptics, *GS Textplorer* by Gurusoft

1.3.4 DM Tool Classification – Cont'd

3) Application-specific tools

- customized environment, *e.g.*, for marketing, churn prediction, customer relations management (CRM)
- strength = provide user with guidance in setting-up data mining project through question-and-answer dialogs
- hence, less expertise from user
- examples: IBM *Intelligent Miner for relationship marketing*, Unica's *Model 1*, SLP Inforware's *Churn/CPS* for churn prediction, Quadstone's *Decisionhouse for CRM*

4) Embedded data mining tools

- added to database management systems (DBMS) products & BI suites
- mostly decision trees only
- easy to use, less specialized staff needed
- restricted flexibility affects quality of results (less accurate), limited functionality for validating results

1.3.4 DM Tool Classification – Cont'd

5) Analytical programming tools

- targeted at generic analytical, tasks, not data mining specifically
- lots of graphics, database access facilities, statistics
- work only for smaller data sets
- requires experienced staff (savvy users)
- examples for **business analyst**: SPSS's *Base*, SQL tools, Excel
- examples for **statistical analyst**: SAS Macro Language, Matlab, S-Plus

6) Data mining solutions & support from external services provider (ESP)

- from advice to development & on-/off-site implementation
- drawback = loss of control by customer, *e.g.*, loss of personnel at ESP → project follow-up?
- project also fail due to bad model selection, wrong business constraints, unknown regulations,. . .
- examples: PriceWaterhouseCoopers (PWC), IBM Global Business Solutions, Data4S,. . .

1.3.4 DM Tool Classification – Cont'd

Bottom line:

- One should know data mining project requirements
  + benefits of all options offered for project
- When data mining objectives are unclear → choose class 1
- **However:** trade-off between: quality of results &
  required skill level, flexibility, time-to-solution
- classes are not *mutually exclusive* (they overlap)
  + can *complement eachother*, in particular for companies with complex or numerous data mining objectives

1.3.4 DM Tool Classification – Cont'd

Evaluation of data mining tool categories:

Class	Ease of deployment	Quality of results	Time-to-solution	Flexibility
Generic	2-3	3-4	2-4	3-4
Alg-spec	2-3	4-5	2-4	2-3
Appl-spec	4-5	2-4	4-5	1-2
Embedded	3-4	1-3	3-5	2-3
Analytical	1-5	2-5	1-5	5
DM solutions	3-5	2-5	2-5	3-5

Ratings: 1 = worst, 5 = best. Ease of deployment is inversely proportional to "in-house skills required"

1.3.5 Overview of Available DM Tools

Commercial- (com) & public domain (pd) systems

- **Classification**
  - Decision-tree approach:
    pd: C4.5, GAtree, IND, Mangrove, OC1, ODBCMINE, PC4.5, SMILES,. . .
    com: AC2, Alice d'Isoft, Angoss KnowledgeSEEKER, Angoss StrategyBUILDER,C5.0 , CART 5.0, DTREG, Decisionhouse, SPSS AnswerTree,Xpertrule Miner,. . .
  - Neural network approach:
    pd: NN FAQ free software list, NuClass7, Sciengy RPF,. . .
    com: Alyuda NeuroIntelligence, BioComp iModel(tm), COGNOS 4Thought, BrainMaker, KINOsuite, MATLAB Neural Net Toolbox, MemBrain, NeuroSolutions, NeuroXL, NeuralWorks Predict, SPSS Neural Connection 2, STATISTICA Neural networks, Synapse, Tiberius, Eudaptics Viscovery, Gurusoft GS Textplorer
  - Rule Discovery approach:
    pd: CBA, DM-II system, KINOsuite-PR, PNC2 Rule Induction System
    *com:*Compumine Rule Discovery System, Datamite, DMT Nuggets, PolyAnalyst, SuperQuery, WizWhy, XpertRule Miner
  - Genetic Programming approach:
    com: Evolver, GAtree, Genalytics GA3, GenIQ Model, Gepsoft GeneXproTools 4.0
  - Other approaches:
    pd: Grobian, Rough Set Exploration System (RSES)
    com: BLIASoft Knowledge Discovery software, Datalogic

1.3.5 Overview Avail. DM Tools – Cont'd

Commercial- (com) & public domain (pd) systems – Cont'd

- **Classification**:
  Decision tree methods
  Rule-based methods
  Neural networks

- **Clustering**
  pd:
  Autoclass C, Databionic ESOM Tools, MCLUST/EMCLUST, PermutMatrix, Snob, SOM in Excel, StarProbe
  com: BayesiaLab, ClustanGraphics3, CViz Cluster Visualization, IBM Intelligent Miner for Data, Neusciences aXi.Kohonen, PolyAnalyst, StarProbe, Viscovery explorative data mining modules, Visipoint

↪ for more information, see:
`http://www.kdnuggets.com/siftware.html`

1.3.5 Overview Avail. DM Tools – Cont'd

Neural Networks in Data Mining:

1. **Additional tool** in *generic data mining suite*, or
   1-of-several algorithms in *algorithm-specific DM tool*

   - (mostly) regression/classification/(time-series) prediction

     Techniques used:

     – *Multilayer Perceptron* (MLP), trained with Backprop

     – *Radial Basis Function* (RBF) networks: 2-layered NN, input layer = RBFs, mostly radially-symm. Gaussians, output layer = 1-layered perceptron

     – *Support Vector Machines* (SVM): optimal choice of classification boundaries by weight vectors (support vectors) also used for regression purposes

     ↪ for more information on RBF, SVM:
     see course Artificial Neural Networks (H02C4A)

     Examples: SPSS' *Clementine*, Thinking Machines' *Darwin*, Right Information Systems *4Thought*, Vienna Univ. Techn. *INSPECT*

   - sometimes (rudimentary) use of *SOM algorithm*
     for clustering & (high-dimensional) data visualization

2. **Prime tool** in *specialized stand-alone tools*
   using topographic maps (SOM) for clustering, high-dimensional data visualization, regression, classification

   ↪ for more information on SOM: see further

   Examples: *Viscovery* by Eudaptics Databionic ESOM Tools *GS Textplorer* by Gurusoft *Neusciences aXi.Kohonen* by Solutions4Planning

1.4 Data Preprocessing

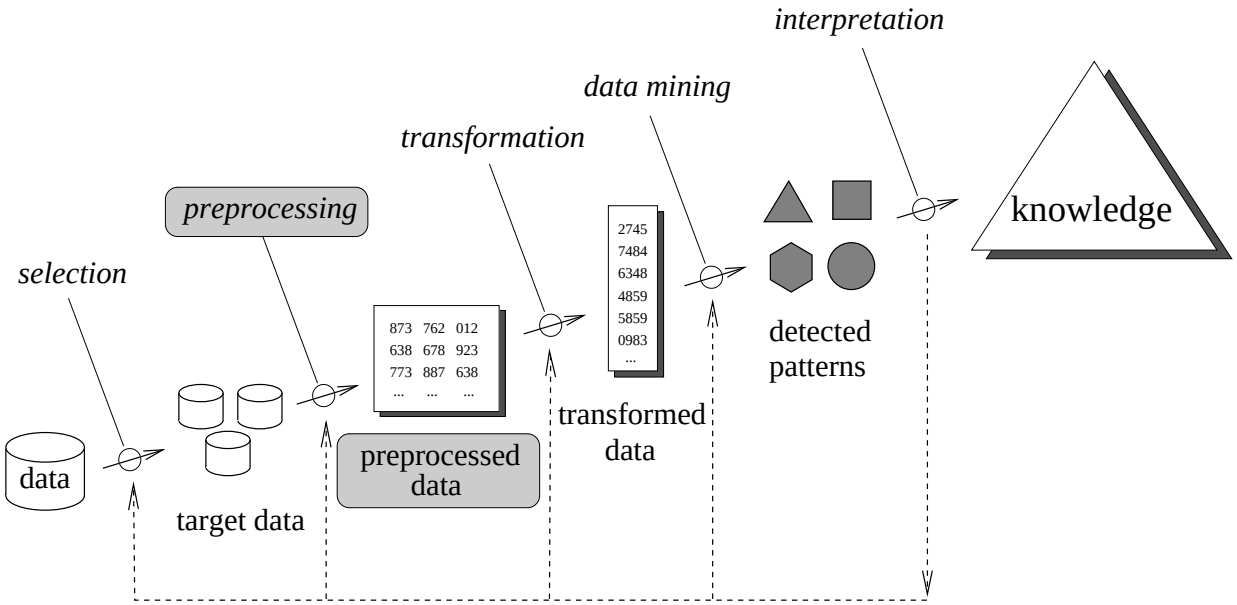
1.4.1 Introduction

- Data Mining is rarely performed on raw data
- **Reasons:**
  1. data can be **noisy**
     e.g.: noisy time series: apply temporal filtering
     underlying idea:
     1) small fluctuations are not important, but trends are
     2) fluctuations can be due to imperfect measuring device
  2. data can contain **outliers/wildshots** (= unlikely samples)
     e.g.: human error in processing poll results, errors filling-out forms by customer in marketing inquiry, etc. . .
     however: outliers could be *nuggets* one is looking for. . .
  3. data can be **incomplete**
     e.g.: not every question in a poll is answered
  4. data can be **unlabeled**
     e.g.: outcome of not every clinical trial is known
  5. **not enough data** to perform *e.g.* clustering analysis
     no clear aggregates observed which could lead to clusters
  6. **too high-dimensional data** to do *e.g.* regression
     not enough data to estimate the many parameters

  ↪ **Hence, prior to DM:**

  - data preprocessing (points 1. . . 4) → this chapter
  - data transformation (points 5 & 6) → next 2 chapters:
    - *project* on subspace/manifold ("Data Transformation")
    - *select* subset of features or inputs ("Feature Selection")

1.4.1 Introduction - Cont'd



Steps in KDD process

Data Preprocessing $\stackrel{\Delta}{=}$ remove noise & outliers, handle missing data & unlabeled data,. . .

- **Outlier removal**
  → distinguish informative from uninformative patterns
- **Noise removal**
  → suppress noise by appropriate filtering
- **Missing data handling**
  → deal with missing entries in tables
- **Unlabeled data handling**
  → deal with missing labels in classification

1.4.2 Outlier removal

- **What is an outlier?**
- **Statistical definition:**
  new pattern $\mathbf{v}_i$ is **outlier** of data set $\mathcal{D}$?
  if probability $P(\mathbf{v}_i \in \mathcal{D}) < 10^{-6}$, *e.g.*
- **Information-theoretic definition:**
  new pattern = **outlier** when difficult to predict by **model** trained on previously seen data
  outlier = **informative pattern**
  - Pattern is informative when it is suprising
  - *Example:* 2 class problem, labels $\in \{0, 1\}$
  - probability that estimated label $\widehat{y_k}$ of new pattern $\mathbf{v}_k$, given classification model, = correct label y_k:

$$\begin{aligned} I(k) &= -\log P(\widehat{y_k} = y_k) = \\ &\quad -y_k \log P(\widehat{y_k} = 1) - (1 - y_k) \log(1 - P(\widehat{y_k} = 1)) \end{aligned}$$

    (Shannon information gain)
  - Information-theoretic sense: pattern $\mathbf{v}_k$ = most informative when $I(k) >$ threshold, else it is uninformative

1.4.2 Outlier removal – Cont'd

1.4.2.1 Data cleaning

- **Garbage** patterns can also be informative!!!
- **Data cleaning**: sort out "good" / "bad" outliers
  sort out "nuggets" from "garbage"
- Purely **manual** cleaning = tedious
- **Hence:** computer-aided tools for data cleaning
  applicable to classification, regression, data density modeling
- **On-line algorithm**:
  1. train model (*classifier*) that provides I estimates on small clean subset
  2. draw labeled pattern $(\mathbf{v}_i, y_i)$ from raw database
  3. check if information gain $I(i) \stackrel{?}{><}$ threshold
     - if $I(i) <$ threshold OK $\Rightarrow$ use for training model
     - if $I(i) >$ threshold $\Rightarrow$ human operator checks: if pattern = garbage ($\rightarrow$ discard) or acceptable ($\rightarrow$ use for training model)
  4. stop when all data has been processed

  Disadvantage: dependence on order patterns are presented

  Question: what is optimal threshold?

1.4.2 Outlier removal – Cont'd

1.4.2.1 Data cleaning – Cont'd

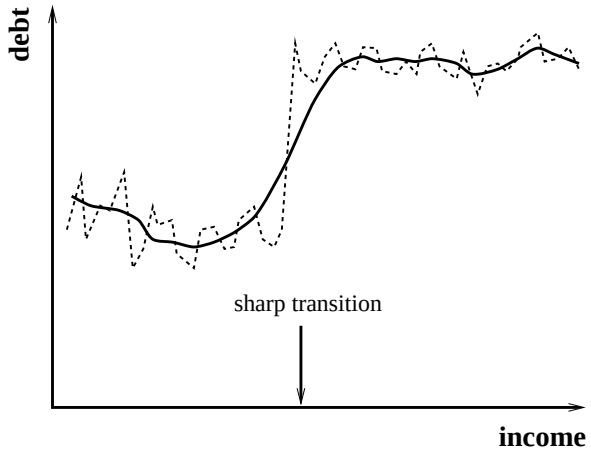
- **Batch algorithm**:
  1. train model on all data (garbage as well)
  2. sort data according to information gain I
  3. human operator checks patterns $\mathbf{v}_i$ with $I(i) >$ threshold remove if garbage
  4. retrain model
  5. sort data according to information gain I
  6. human operator removes garbage patterns
  7. . . .

  Question: what is optimal threshold?

- **Optimal** threshold:
  1. perform data cleaning several times, for different thresholds *i.e.*, for series of increasing threshold values
  2. determine model errors on test set (validaton error)
  3. choose model for which validaton error is lowest
     $\Rightarrow$ optimal threshold

1.4.3 Noise Removal

1. *lowpass filtering* = linear filtering
   principle: replace signal value at time t, $s[t]$, by weighted average of signal values in Gaussian region around t
   example: $s[t] \leftarrow 0.5s[t] + 0.25s[t-1] + 0.25s[t+1]$
   advantage: removes noise when cut-off frequency Gaussian filter $<$ expected lowest frequency in noise
   disadvantage: blurs sharp transitions



2. *median filtering*

3. *regression*

1.4.3 Noise Removal – Cont'd

1. *lowpass filtering*

2. *median filtering* = non-linear filtering
   principle: replace signal value at time t, $s[t]$,
   by *median* of signal value in region $\mathcal{R}$ around t
   advantages:

   - less blurring than lowpass filtering
   - very effective if noise consists spike-like components ("outliers")

   $\hookrightarrow$ often better suited for noise removal than lowpass filt.

3. *regression* (curve fitting): signal can be fitted by polynomial or parametric function with small enough # of parameters
   How? *cf.*, generalization performance in NN training
   vs. network complexity (*i.e.*, # weights)

4. . . .

1.4.4 Missing Data Handling

1. *Mean substitution* = most used technique
   How?: replace missing entries in column by column's mean
   $\hookrightarrow$ crude but easy to implement

2. *Cluster center substution*
   look for nearest cluster center μ_c, by ignoring missing entries & substitute (Kohonen, 1995; Samad & Harp, 1992)

3. *Expectation Maximization* (EM) technique
   more sophisticated technique (Dempster *et al.*, 1977)
   statistics-based: replace missing entries by most likely value

4. . . .

1.4.5 Unlabeled Data Handling

Two basic strategies:

1. discard unlabeled entries

2. use unlabeled + labeled entries & model data density
   then use density model to develop classification model
   (Hence, in this way, <u>all</u> data is used as much as possible)