

---

# KNOWLEDGE MANAGEMENT & BUSINESS INTELLIGENCE

---

*Course notes*



2016-2017

KUL

Ysaline de Wouters

## Table of content

<b>CHAPTER 1 : INTRODUCTION</b>	<b>6</b>
<b>1. BASIC CONCEPTS: KNOWLEDGE AND INTELLIGENCE</b>	<b>6</b>
1.1 THE DIKW HIERARCHY	6
1.2 KNOWLEDGE	7
<b>2. BUSINESS INTELLIGENCE AND ARTIFICIAL INTELLIGENCE</b>	<b>8</b>
<b>CHAPTER 2: ENTERPRISE BUSINESS INTELLIGENCE</b>	<b>11</b>
<b>1. BI OVERVIEW</b>	<b>11</b>
<b>2. INFORMATION OVERLOAD</b>	<b>11</b>
<b>3. BI OBJECTIVES AND BENEFITS</b>	<b>13</b>
3.1 BI OBJECTIVES	13
3.2 PERSPECTIVES IN BUSINESS INTELLIGENCE	13
3.3 BI VIEWS	14
3.4 GOALS OF BUSINESS INTELLIGENCE	15
<b>CHAPTER 3: DATA WAREHOUSING</b>	<b>16</b>
<b>1. DATA INTEGRATION</b>	<b>16</b>
<b>2. DATA WAREHOUSING</b>	<b>16</b>
<b>3. ARCHITECTURE</b>	<b>18</b>
3.1 STAR SCHEMA	18
3.2 SNOWFLAKE SCHEMA	18
<b>4. DATA EXTRACTION</b>	<b>19</b>
<b>5. TABLE FORMATS AND ONLINE ANALYTICAL PROCESSING (OLAP)</b>	<b>21</b>
<b>6. LOG FORMATS</b>	<b>22</b>
<b>7. FROM TRANSACTIONAL DATA TOWARDS ANALYTICAL DATA</b>	<b>23</b>
<b>8. REPORTING &amp; ANALYSIS</b>	<b>23</b>
<b>9. APPLICATIONS</b>	<b>25</b>
<b>CHAPTER 4: KNOWLEDGE DISCOVERY IN DATA (KDD)</b>	<b>27</b>
<b>1. WHAT IS KDD?</b>	<b>27</b>
1.1 OVERVIEW	27
1.2 STEPS IN THE KDD PROCESS	28
<b>2. DATA PRE-PROCESSING</b>	<b>28</b>
<b>3. DATA MINING TASKS AND TECHNIQUES</b>	<b>29</b>
<b>3.1 PREDICTIVE TECHNIQUES</b>	<b>29</b>
3.1.1 CLASSIFICATION	29
3.1.2 REGRESSION	30
<b>3.2 DESCRIPTIVE TECHNIQUES</b>	<b>34</b>
3.2.1 ASSOCIATION RULES	34
3.2.2 SEQUENCES	35
3.2.3 WEB MINING	36
3.2.4 CLUSTERING	36
<b>4. POST-PROCESSING</b>	<b>37</b>
<b>5. APPLICATIONS</b>	<b>37</b>
5.1 CREDIT SCORING	37
5.2 FRAUD DETECTION	38
<b>6. ISSUES AND CONCLUSIONS</b>	<b>39</b>



<b>CHAPTER 5: DECISION TREE INDUCTION</b>	<b>41</b>
1. INTRODUCTION: LEARNING TASKS	41
2. METHOD AND TERMINOLOGY	42
3. ADVANTAGES AND DISADVANTAGES OF DECISION TREES	46
<b>CHAPTER 6: KNOWLEDGE BASED SYSTEMS</b>	<b>48</b>
1. KBS - AN OVERVIEW	48
2. REASONING	50
2.1 FORWARD CHAINING (USED FOR PLANNING)	50
2.2 BACKWARD CHAINING (USED FOR DIAGNOSIS)	51
3. MAIN KNOWLEDGE REPRESENTATION FORMS	51
<b>CHAPTER 7: REGRESSION</b>	<b>55</b>
1. MODEL FORMULATION AND TERMINOLOGY	55
2. LINEAR REGRESSION	55
3. LOGISTIC REGRESSION	56
<b>CHAPTER 8: CLUSTERING</b>	<b>57</b>
1. UNSUPERVISED DATA MINING	57
2. CLUSTERING	57
2.1 HIERARCHICAL CLUSTERING	57
2.2 PARTITIONAL CLUSTERING	60
2.3 MODEL-BASED CLUSTERING	62
3. NUMBER OF CLUSTERS	62
4. APPLICATIONS	62
5. VALIDATION	63
<b>CHAPTER 9: KNOWLEDGE REPRESENTATION &amp; REASONING</b>	<b>64</b>
1. FRAME BASED SYSTEMS	64
1.1 FRAMES	64
1.2 INHERITANCE	64
1.3 FACETS	65
2. FUZZY SETS AND REASONING	65
3. INFERENCE AND REASONING	66
3.1 RECALL	67
3.2 SEARCHING THE GOAL STATE SPACE	68
3.3 THE INFERENCE CYCLE	70
<b>CHAPTER 10: DECISION ANALYTICS</b>	<b>73</b>
1. FRAMING ANALYTICS DECISIONS	73
2. DECISION MANAGEMENT	75
3. DECISION MINING	79
<b>CHAPTER 11: PRE-PROCESSING</b>	<b>83</b>
1. DATA GENERATION	83
2. PRE-PROCESSING	83
2.1 DATA EXPLORATION	83
2.2 PRE-PROCESSING	84
2.3 TECHNIQUE SELECTION	85

<b>3. WHICH FEATURES TO USE FOR PREDICTIVE MODELS?</b>	<b>85</b>
<b>4. RELATED TOPICS</b>	<b>89</b>
4.1 DATA QUALITY	89
4.2 IMPUTATION TECHNIQUES	91
4.3 OUTLIERS	91
<b>CHAPTER 12: EVALUATION</b>	<b>92</b>
<b>1. VALIDATION</b>	<b>92</b>
<b>2. PERFORMANCE METRICS</b>	<b>92</b>
<b>CHAPTER 13: DATA VISUALIZATION</b>	<b>94</b>
<b>1. INTRODUCTION</b>	<b>94</b>
<b>2. DATA VISUALIZATION PRINCIPLES</b>	<b>95</b>
<b>3. BEST PRACTICES</b>	<b>96</b>
<b>CHAPTER 14: UNCERTAINTY MODELING</b>	<b>100</b>
<b>1. INTRODUCTION</b>	<b>100</b>
<b>2. APPROACHES TO DEAL WITH UNCERTAINTY</b>	<b>100</b>
2.1 PROBABILITY THEORY	100
2.2 LUKASIEWICZ MULTI-VALUED LOGIC	101
2.3 HEISENBERG PRINCIPLE	101
2.4 FUZZY SET	101
2.5 THEORY OF EVIDENCE	105
2.6 CERTAINTY FACTOR THEORY	108
2.7 POSSIBILITY THEORY	110
<b>CHAPTER 15: KNOWLEDGE DISCOVER IN DATA</b>	<b>111</b>
<b>1. ASSOCIATION RULE MINING</b>	<b>111</b>
1.1 BASIC SEMANTICS	111
1.2 MINING ASSOCIATION RULES	112
1.3 A-PRIORI ALGORITHM	113
<b>2. SEQUENCE PATTERN MINING</b>	<b>115</b>
<b>CHAPTER 16: KNOWLEDGE MANAGEMENT</b>	<b>118</b>
<b>1. INTRODUCTION TO KNOWLEDGE MANAGEMENT</b>	<b>118</b>
<b>2. DIKW</b>	<b>118</b>
<b>3. KM FRAMEWORKS</b>	<b>120</b>
<b>4. KM TOOLS AND TECHNIQUES</b>	<b>122</b>
4.1 NON-ICT PLATFORMS FOR KNOWLEDGE MANAGEMENT	122
4.2 ICT PLATFORMS FOR KM	124
<b>5. ROLE OF ICT FOR KM</b>	<b>124</b>
<b>CHAPTER 17: KNOWLEDGE BASED SYSTEMS</b>	<b>127</b>
<b>1. INTERFACES</b>	<b>127</b>
1.1 USER INTERFACE	127
1.2 SYSTEM INTERFACE	128
1.3 DEVELOPER INTERFACE	128
<b>2. VERIFICATION AND VALIDATION</b>	<b>129</b>
<b>CHAPTER 18: DECISIONS, RULES AND PROCESSES</b>	<b>135</b>

<b>1. INTRODUCTION</b>	<b>135</b>
<b>2. DECISION MODEL &amp; NOTATION</b>	<b>135</b>
2.1 PRIMARY USE CASES	136
2.2 DMN LEVELS	136
2.3 DECISION REQUIREMENT GRAPH	136
2.4 DECISION TABLE	137
<b>3. DECISIONS AND PROCESSES: THE ROLE OF DECISION MODELS</b>	<b>137</b>
<b>4. EXECUTION SCENARIOS</b>	<b>139</b>
4.1 EXECUTION SCENARIO 1	139
4.2 EXECUTION SCENARIO 2	140
4.3 EXECUTION SCENARIO 3	141
<b>CHAPTER 18: ARTIFICIAL NEURAL NETWORKS AND SUPPORT VECTOR MACHINES</b>	<b>142</b>
<b>1. NEURAL NETWORK</b>	<b>142</b>
1.1 CHARACTERISTICS	142
1.2 THE NEURON MODEL	143
1.3 LEARNING WITH PERCEPTRON	144
1.4 BACKPROPAGATION LEARNING	146
1.5 LEARNING VS. MEMORISATION	148
<b>2. MOTIVATION SVM</b>	<b>151</b>
<b>CHAPTER 19: ENSEMBLE LEARNERS, SURVIVAL ANALYSIS, SOCIAL NETWORK ANALYTICS, ORGANIZATIONAL ASPECTS OF ANALYTICS</b>	<b>156</b>
<b>1. ENSEMBLE LEARNERS</b>	<b>156</b>
1.1 BAGGING	156
1.2 BOOSTING	157
1.3 STACKING	158
1.4 RANDOM FOREST	158
<b>2. SURVIVAL ANALYSIS</b>	<b>159</b>
<b>3. SOCIAL NETWORK ANALYTICS</b>	<b>159</b>
3.1 NETWORKS EXPLAINED	160
3.2 HOMOPHILY EXPLAINED	161
3.3 CASE STUDY: CHURN PREDICTION IN TELCO	161
<b>4. ORGANIZATIONAL ASPECT OF ANALYTICS</b>	<b>162</b>
4.1 HOW TO ORGANIZE YOUR DATA SCIENCE TEAM?	162
4.2 EXPERIMENTAL STUDY	164

## CHAPTER 1 : Introduction

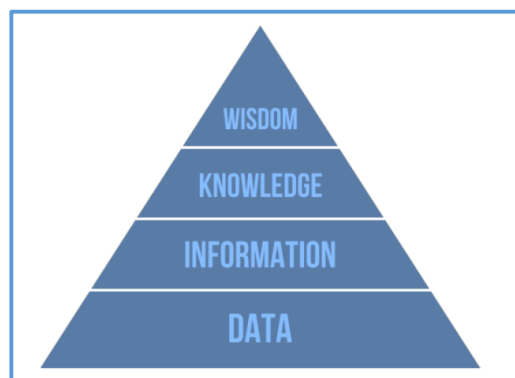
### 1. Basic concepts: knowledge and intelligence

⇒ This course is about **knowledge and intelligence**

#### 1.1 The DIKW hierarchy

- How to build smart systems that use knowledge
- How to organize knowledge
- How to ensure knowledge quality
- How to discover knowledge from data
- How to get value out of it
- How to make it actionable
- How to manage knowledge
- How to make decisions based on knowledge

The **DIKW hierarchy** is used for representing structural and or functional relationships between 4 elements : Data, Information, Knowledge, Wisdom.



- **Data...** data is raw. Raw facts, numbers, documents. Data do not have an intrinsic value, but receive it in a certain context and for a specific audience. It simply exists and has no significance beyond its existence. It can exist in any form, usable or not. It does not have meaning of itself.
- **Information:** Data that has been given meaning by way of relational connection. Information is relevant data made available on time and in the correct form. Management decisions can be taken, based upon this information.
- **Knowledge** is the appropriate collection of information, such that it is intent to be useful. Knowledge refers to the way to deal with the information, to take decisions, to establish relations between data. This does not refer to factual knowledge but knowledge to deal with the facts.
- **Wisdom** is an extrapolative and non-deterministic, non-probabilistic process. It calls upon all the previous levels of consciousness, and specifically upon special types of human programming (moral, ethical codes, etc.). Unlike the previous

four levels, it asks questions to which there is no (easily-achievable) answer, and in some cases, to which there can be no humanly-known answer period. Wisdom is therefore, the process by which we also discern, or judge, between right and wrong, good and bad. Wisdom is a uniquely human state.

## 1.2 Knowledge

Knowledge is present in **various locations**: in the head of individuals, in the collective memory of an organization, written down in texts, procedures, manuals, hidden in information systems, etc.

### Types of knowledge

- **Explicit knowledge**: written down or easy to register
- **Tacit knowledge**: unconscious, unspoken, subjective or heuristic knowledge

The challenge is to

- Detect: knowledge discovery
- Represent: knowledge representation
- Manage: knowledge management
- Distribute: knowledge distribution
- Apply to relevant problem situations this knowledge

### Applicable in multiple areas

- OLAP, dashboards, visualisations
- Smart systems, sensors, cities, cars
- Business analytics, Knowledge discovery, data mining, machine learning, deep learning
- Artificial intelligence, Knowledge Based systems, Fuzzy systems
- Virtual assistants

### Overflow of data

The extensive use of social media, sensor applications, as well as the immediate provision of possibly large result data by modern search engines has led to a massive increase in produced and potentially interesting-to-analyse data.

### Four V's

- **Large volume** of data: refers to processing huge amounts of data
  - Many objects: customers, products, etc.
  - Many features: age, income, etc.
- **Velocity**: Refers to the frequency with which new data enters the integration and analysis process. Data stream. Ex: phone calls, bank transactions, web site visits
- **Variety**: different types of "data". Ex: numbers, text, speech, video, audio, scanned documents, photos, social media updates, sensor data.

- **Veracity:** messy, untrustworthy data

Facing big data, the new challenges for the ETL process are mainly caused by data velocity, i.e., just-in-time data extraction becomes necessary.

Shift from ETL to ELT has been suggested for big data projects. The idea is that data storage frameworks work on the raw data before making it available to other systems for further analysis.

Transformation and cleaning steps are shifted after first analyses have been applied. In this way, analyses can be conducted more rapidly, shifting expensive transformation and cleaning tasks to later phases of the project.

## 2. Business Intelligence and Artificial Intelligence

There exist **many different** definitions for the business intelligence term.

- Business intelligence can be defined as the ability to collect and react accordingly based on the information retrieved
- The ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal
- The set of techniques and tools for the transformation of raw data into meaningful and useful information for business analysis purposes
- Tools and systems that play a key role in the strategic planning process of the corporation. These systems allow a company to gather, store, access and analyse corporate data to aid in decision-making

Applications: Generally, these systems will illustrate business intelligence in the areas of customer profiling, customer support, market research, market segmentation, product profitability, statistical analysis, and inventory and distribution analysis to name a few.

### 3 core meanings

- 1) **Intelligent & Knowledge Based systems:** Knowledge representation, reasoning, Business rules & decisions, smart processes, Knowledge management.
- 2) **Enterprise reporting:** Data warehouses, OLAP, Corporate Performance Management, dashboards.
- 3) **Knowledge discovery & Business Analytics:** data mining, process mining, web mining, opinion mining, decision mining.

### BI questions

- What happened?
- What is happening?
- Why is it happening?
- What will happen?
- What do I want to happen?

## Knowledge Based systems

An intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution.

### 2 aspects

- **Functional aspect:** real problems, performance (effectiveness & efficiency), expertise
- **Internal aspect:** knowledge base, inference

Why knowledge based systems

Pros	But
<ul style="list-style-type: none"> <li>• Knowledge is permanent</li> <li>• Available at any time</li> <li>• Consistent and reproducible decision making</li> <li>• No constant rehearsal necessary</li> <li>• Easy and cheap to transfer</li> <li>• Easy to document (training)</li> </ul>	<ul style="list-style-type: none"> <li>• Creativity, flexibility, evolution?</li> <li>• Common sense knowledge?</li> <li>• Broad focus?</li> <li>• Simultaneous reasoning</li> <li>• Responsibility, legal?</li> </ul>

## Wrap-up artificial intelligence

- AI is the study of mental faculties through the use of computational models.
  - At bottom, AI is about the simulation of human behaviour.
  - AI is the discipline that aims to understand the nature of human intelligence through the construction of computer programs that imitate intelligent behaviour
  - AI is the study of how to make computers do things at which, at the moment, people are better
  - AI is the part of computer science concerned with designing intelligent computer systems, i.e. systems that exhibit the characteristics we associate with intelligence in human behaviour: understanding language, learning reasoning, solving problems.
- ⇒ What is Artificial Intelligence? Making computers think? Building intelligent machines and applications? Using computers to study human intelligence?
- ⇒ What is intelligence? Everything a human can do better?

**AI paradox:** AI is not about unusual clever insights but about ordinary skills and people.

But, what is easy and what is hard? It seems **easy** to automate high level tasks we usually associate with human “intelligence”. Ex: mathematical proofs, chess, medical diagnosis, etc. It seems **hard** to automate some basic skills: walking

through a room with obstacles, driving, interpreting visual information, recognizing faces.

→ What is the fundamental difference?

**Characteristics of AI programs**

- Symbolic representation and processing
- Heuristics: no general algorithm known
- Knowledge representation
- Incomplete and conflicting data
- Ability to learn
- Ability to explain



## CHAPTER 2: ENTERPRISE BUSINESS INTELLIGENCE

### 1. BI Overview

The term **Business Intelligence** represents the tools and systems that play a key role in the strategic planning process of the corporation. These systems allow a company to gather, store, access and analyse corporate data to aid in decision-making. Generally, these systems will illustrate business intelligence in the areas of customer profiling, customer support, market research, market segmentation, product profitability, statistical analysis, etc.

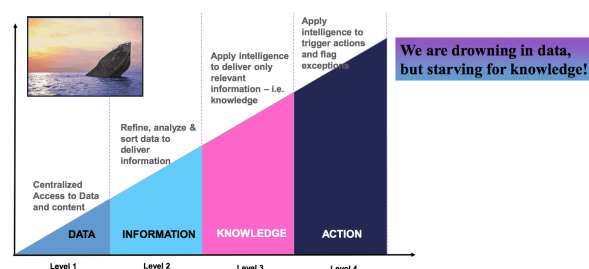
⇒ Business Intelligence is the process of gathering meaningful information to answer questions and identify significant trends or patterns, giving key stakeholders the ability to make better business decisions.

In the last years, data availability and analysis capabilities have increased tremendously, and new research areas for BI have emerged.

- Tasks of BI: well-structured understanding of the business logic.  
New organizational structures like decentralized organizations want to apply decisions support within their environment and, hence, ideas from collective intelligence or crowds sourcing are applied in BI.
- Foundations of BI: datawarehouse + data on the web. Such data is often not well-structured, but only semi structured such as text data.  
Need for integration has led to models for linking data in BI. Consequently, scope has broadened and new tools such as visual mining, text mining, opinion mining, ... have emerged.
- Realization of BI systems: from computational point of view, we have to deal with large and complex data sets nowadays.
- Delivery of BI: mobile devices offer a new dimension for delivering information to user sin real-time.

Problem: we have so much data in a company, we want to integrate it. Information is available but in many different places ...

### 2. Information overload



We have too much information and we try to obtain knowledge from this information and improve decision making.

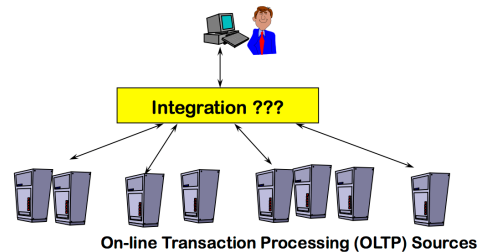
- Data = Representations of reality
- Information = Data which provides relevant clues or news
- Knowledge = The framework or schema for organizing the relationships between pieces of information.
- Action = The deeds or decisions made based on knowledge

Smart decisions: EDM (Enterprise decision management)

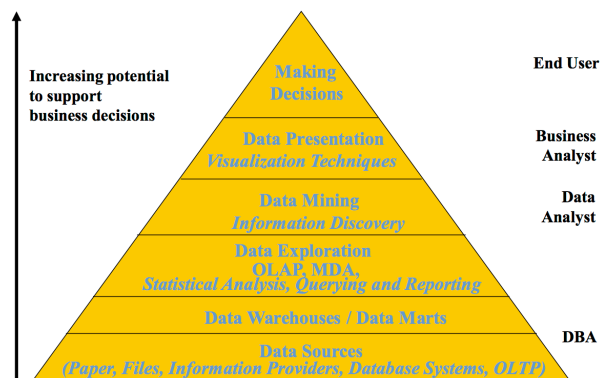
- Modelling, managing and executing the business decisions
- Descriptive, Predictive, Prescriptive Analytics, BI
- CEP: Complex Event Processing

Problem: Integration. A lot of data and information is available from many sources. Hence, we need to integrate this data.

- Collect and combine information
- Provide integrated view, uniform user interface
- Support sharing



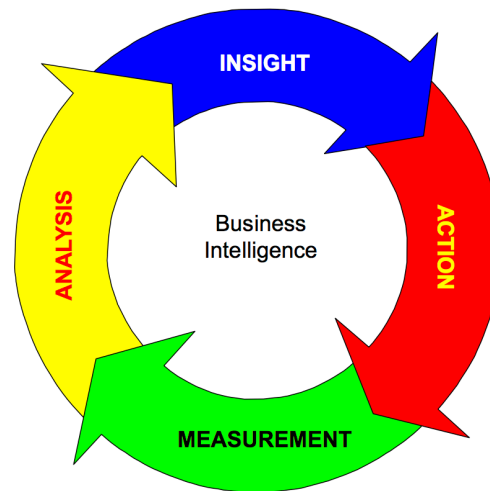
The problem becomes even more complex when we try to play with these data sources.



### Why Business Intelligence?

- Automated data collection tools and mature database technology lead to tremendous amounts of data stored in databases (legacy data, ERP, scanner data, documents, RFID, mobile, etc.)
- Business intelligence refers to an interactive process for exploring, analysing and reporting structured, domain-specific data, thereby deriving insights and drawing conclusions.
- Business intelligence systems are developed to support strategic and tactical decisions and to assess business performance more precisely.

### The Business Intelligence cycle



#### Ineffective approaches

⇒ Why can't we just put data in spreadsheets or in a small database?

- Simple approaches are not enough
- Local databases, SQL, simple spreadsheets. Problems: consistency, security, poor user productivity; limited capabilities, one database only
- ERP/CRM reporting and analysis. Problems: one source only. It remains too complicated to put it in only one tool. We need something more sophisticated.
- Isolated tools. Problems: diverse user needs; multitude of tools; TCO
- Home solutions. Problems: maintenance and scaling

### 3. BI Objectives and benefits

#### 3.1 BI objectives

- **Flexibility & Agility**: real-time business information gives companies the capability to react more quickly to changing business conditions and to make necessary adjustments much earlier than in the past.
- **Legal obligations**: Sarbanes-Oxley Act, Basel II. Certifying that, to the best of knowledge, the quarterly and annual filings do not contain any false data or omission of facts and correctly represent the financial situation of the company
- **Responsibility and accountability**: analysing important metrics, improve efficiency, clearer cost allocation.

#### 3.2 Perspectives in Business Intelligence

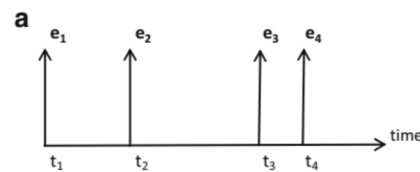
- **Production perspective**: what kind of products should be offered to the customers and how should the production be operated?  
Important role for product development and for internal organization of the business.

- **Customer perspective:** focus on customer behaviour and aims at understanding how customers perceive products or services and how they react to this offer. Essential role in service-oriented business.
- **Organizational perspective:** examines organizational back-ground of the business process.

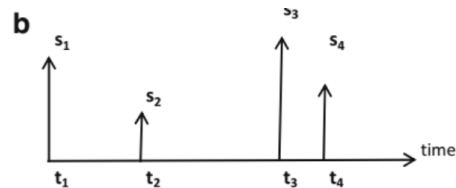
### 3.3 BI views

**Event view:** The main emphasis is on the events in the business process characterized by a time stamp for the start, a time stamp for the end, and, if necessary, also a time stamp for the resumption of the activity execution after an interruption.

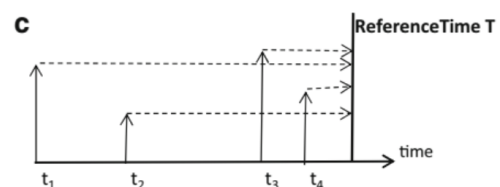
Problem: not always easy to record the exact start and end event. Ex: illness

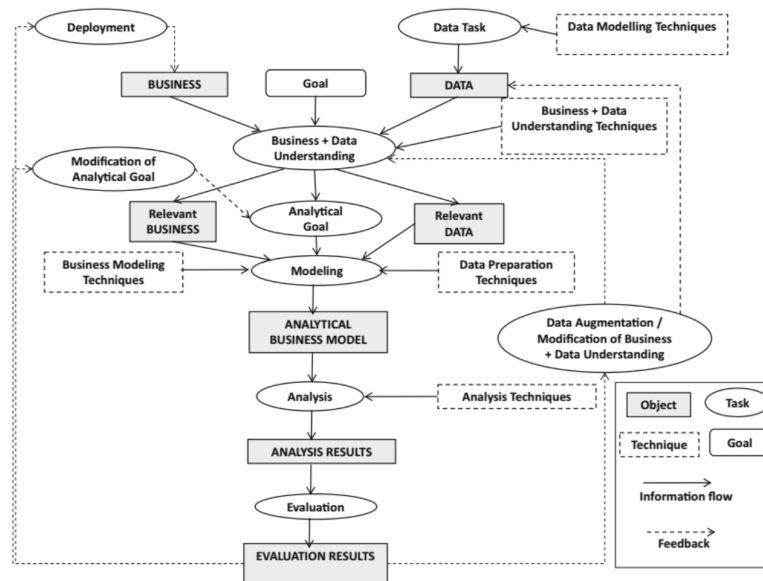


**State view:** Besides the occurrence of events the state view also considers the values of attributes, the so-called state variables, measured in connection with the events.



**Cross sectional view:** In this case, we investigate the history of many process instances at a certain reference time. Usually, this view considers information about events as well as the values of state variables and summarizes the information about process instances for decision making.





The iMine method starts from the available data that has been provided by the data task employing data modelling techniques. Data and business together with a certain analysis goal provide the input for the task of business and data understanding. Thereby, business and goal together represent that part of reality for which we aim for decision support. The output of this task are analytical goals together with relevant business and relevant data, possibly as a certain fragment or view on the original business and data along the different BI perspectives customer, production and organization.

Based on the results from business and data understanding, an analytical business model is built in the modelling task using business modelling techniques and data preparation techniques.

The analytical business model is then analysed by different analysis techniques, leading to analysis results. These results are then evaluated yielding evaluation results.

### 3.4 Goals of Business Intelligence

⇒ Analysis goals: range from the **acquisition of information** about some aspects of the business process over **improving the performance** of the process up to **understanding the implications** of the process for achieving strategic goals.

2 Different ways:

- **Key performance indicators**: allow measuring the performance of the business with respect to goals in any perspective of the business.  
Ex: economic context, KPI may refer to acquisition of new customers, drop-out of students,
- **Analytical goals**
  - Descriptive goals
  - Predictive goals
  - Understanding goals

## CHAPTER 3: DATA WAREHOUSING

### 1. Data Integration

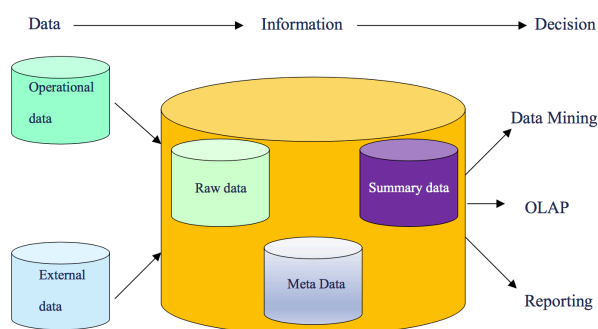
We build a database which is a copy of all the data we have. Since memory is cheap, it is easy and not expensive to copy huge amounts of data.

#### The Objectives

- Data has been collected everywhere and in huge amounts. How to make good use of your data?
- Bring together scattered information from multiple sources as to provide a consistent database source from decision support queries.
- Off-load decision support applications from the on-line transaction system
- Provide architectures and tools for business executives to systematically organize, understand, and use their data to make strategic decisions.

⇒ Of course, when integrating data, we do it in an intelligent manner and we don't copy everything.

### 2. Data Warehousing



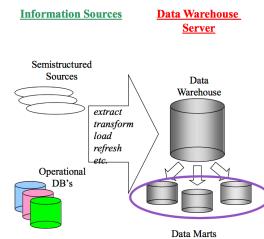
We fill the data warehouse with all kind of data (external, internal, etc.). once this is done, data is available for reporting.

A data warehouse is subject-oriented, integrated, time-varying, non-volatile collection of data that is used primarily in organizational decision making.

- **Subject-oriented:** data is analysed according to what we want to do. Data warehouses are designed to help you analyse data. For example, to learn more about your company's sales data, you can build a warehouse that concentrates on sales. Using this warehouse, you can answer questions like "Who was our best customer for this item last year?"
- **Integrated:** Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as naming conflicts and inconsistencies

among units of measure. When they achieve this, they are said to be integrated.

- **Time-varying:** we only add information
- **Non-volatile:** we never delete information, we only update. In order to discover trends in business, analysts need large amounts of data. This is very much in contrast to online transaction processing (OLTP) systems, where performance requirements demand that historical data be moved to an archive. A data warehouse's focus on change over time is what is meant by the term time variant.

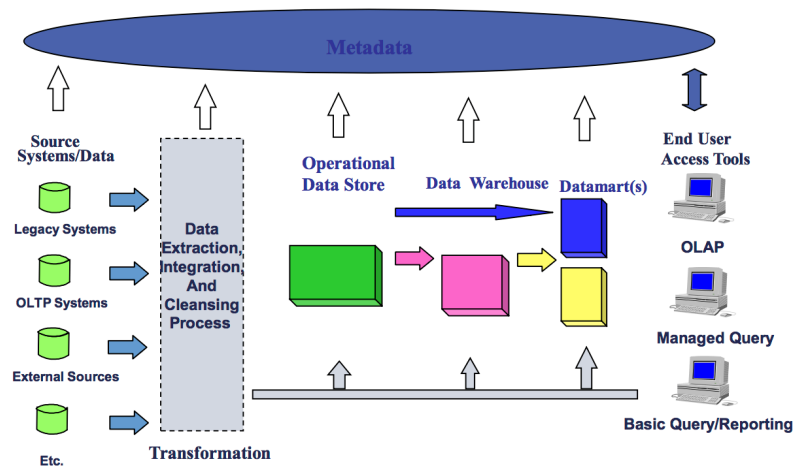


To be distinguished

- 1) **Enterprise warehouse:** collects all information about subjects (customers, products, sales, assets, personnel) that span the entire organization. It requires extensive business modelling (may take years to design and build).
- 2) **Data Marts:** Departmental subsets that focus on selected subjects.
  - Marketing data mart: customer, product, sales
  - Faster roll out, but complex integration in the long run
- 3) **Virtual warehouse:** views over operational DBs.
  - Materialize selected summary views for efficient query processing
  - Easy to build but require excess capability on operational DB servers

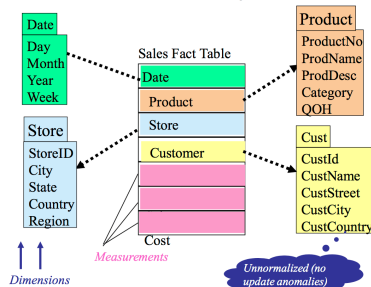
There are some **issues** related to data warehousing

- Modelling a data warehouse: how to organize and model it?
- Building a data warehouse architecture
- How to get information into warehouse: Extract – Transform – Load
- What to do with data once it is in warehouse?
  - Exploring
  - Mining



### 3. Architecture

#### 3.1 Star schema

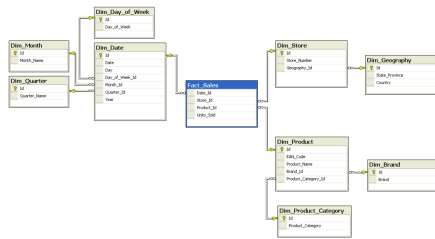


The star schema architecture is the simplest data warehouse schema. It is called a star schema because the diagram resembles a star, with points radiating from a centre. The centre of the star consists of **fact table** and the points of the star are the **dimension tables**. Usually the fact tables in a star schema are in third normal form (3NF) whereas dimensional tables are de-normalized. Despite the fact that the star schema is the simplest architecture, it is most commonly used nowadays.

- **Fact Tables:** A fact table typically has two types of columns: foreign keys to dimension tables and measures those that contain numeric facts. A fact table can contain fact's data on detail or aggregated level.
- **Dimension Tables:** A dimension is a structure usually composed of one or more hierarchies that categorizes data. If a dimension hasn't got a hierarchies and levels it is called flat dimension or list. The primary keys of each of the dimension tables are part of the composite primary key of the fact table. Dimensional attributes help to describe the dimensional value.  
→ Only one table per dimension that holds all information on this dimension.

#### 3.2 Snowflake Schema

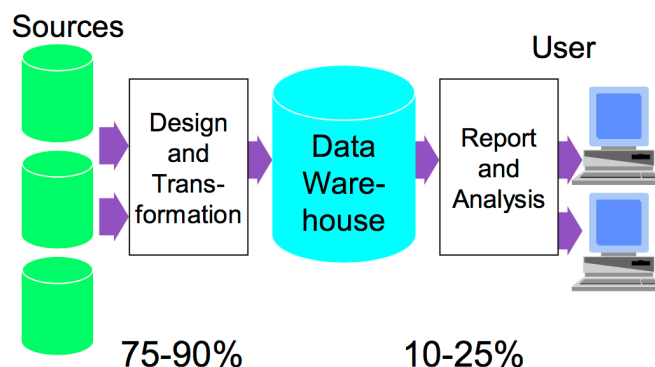




A snowflake schema is a logical arrangement of tables in a multidimensional database such that the entity relationship diagram resembles a snowflake shape. The snowflake schema is represented by centralized fact tables which are connected to multiple dimensions "Snowflaking" is a method of normalising the dimension tables in a star schema.

The snowflake schema is similar to the star schema. However, in the snowflake schema, dimensions are normalized into multiple related tables, whereas the star schema's dimensions are denormalized with each dimension represented by a single table. A complex snowflake shape emerges when the dimensions of a snowflake schema are elaborate, having multiple levels of relationships, and the child tables have multiple parent tables ("forks in the road").

- ⇒ Both schemata have in common that they organize facts and dimensional data within different tables, i.e., there is a separate fact table and tables that represent the dimensions.
- ⇒ Star schema provides a more compact representation of multidimensional data. This comes at the cost of de-normalization. The snowflake schema constitutes a normalization of the star schema.

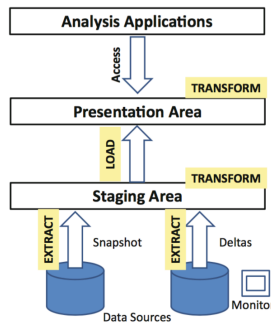


#### 4. Data Extraction

After selecting the relevant data sources and describing them including analysis questions and formats, the next step is to extract relevant data from their sources.

- ⇒ Data extraction is part of the so-called extraction-transformation-load (ETL-process).

#### ETL process

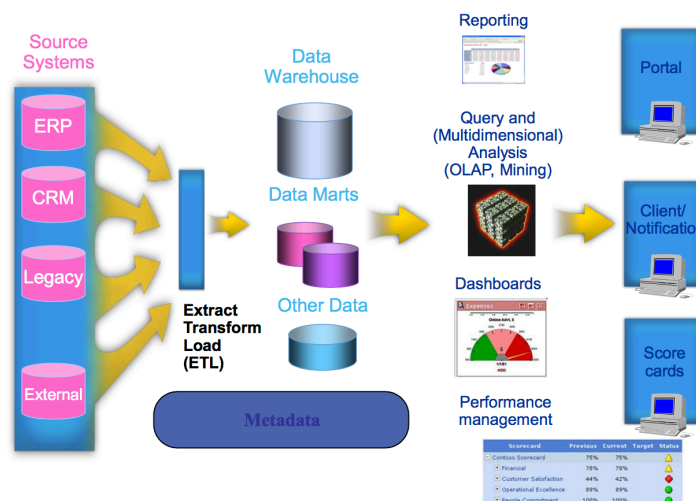


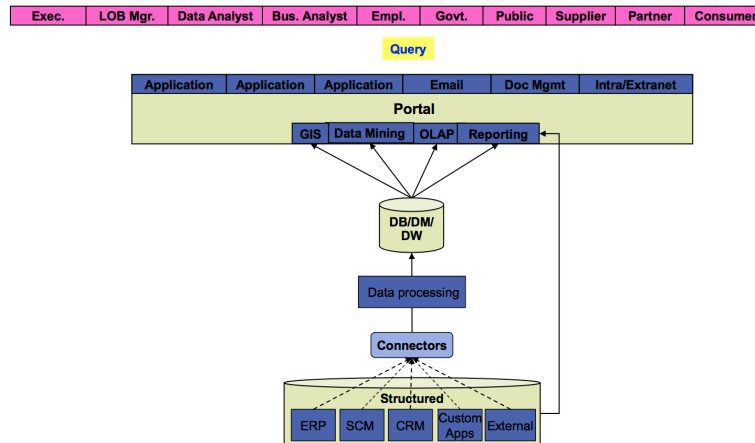
Data stored within different source systems is extracted into the so-called staging area, that provides different services for transforming and cleaning data.

From the staging area, the cleaned and integrated data can be loaded into a presentation area where users can perform analyses.

## Data Warehouse Loading

- **Data extraction/consolidation:** get data from multiple, heterogeneous, and external sources.
  - When to extract?
  - What to extract? Only extract the delta compared to the last data snapshot within the staging area
  - How to extract? SQL queries, database logs.
- **Data cleaning:** detect errors/duplications in the data and rectify them when possible (KUL, K.U.L., K.U.Leuven)
- **Data transformation:** convert data from legacy or host format to warehouse format (Male = M, 1, true)
- **Load:** sort, summarize, consolidate, compute views, check integrity, and build indices and partitions
- **Refresh:** propagate the updates from the data sources to the warehouse (daily 66%, weekly 20%)





Verification driven “intelligence”

## 5. Table formats and Online Analytical Processing (OLAP)

- Query and reporting
  - You know exactly what you are looking for
  - SQL
- OLAP
  - Advanced query and reporting
  - Multidimensional analysis: aggregation (roll-up), drill-down, slice
  - Nice visualisation, data cubes, slice and dice, ...
  - Simulation and What-if scenarios
- Ad hoc analysis and Statistical verification techniques

Table formats refers to multidimensional table structures. As a common metaphor, the data cube is used.

Instead of only looking at the sheer facts, it is often more interesting to analyse them in a different context. This context is realized by different properties or dimensions that describe the facts.

→ Different granularity levels within the dimensions are reflected within the multidimensional model. This enables operations, such as roll up or drill down.

Ex: Month → Quarter → Year

Operations that change the granularity of the data

- **Aggregation**
- **Roll up:** generates new information by the aggregation of data along the dimension where the number of dimensions is not changed. Ex: from Quarter to Months
- **Drill down:** navigation from aggregated data to detailed data.

Operations that don't change the granularity of the data

- **Slice:** generates individual views by cutting “slices” from the cube. In general, this reduces the number of dimensions. Ex: interested in the average duration of all process instances for a certain patient X in the current year.
- **Dicing:** the dimensionality of the cube is not reduced, but the cube itself is reduced by cutting out a partial cube. Ex: number of patient treatments within a certain time frame.

### OLAP Types

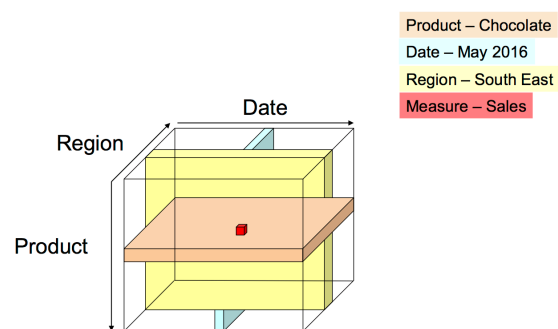
OLAP is an acronym for Online Analytical Processing. OLAP performs multidimensional analysis of business data and provides the capability for complex calculations, trend analysis, and sophisticated data modelling. I

- MOLAP: multidimensional OLAP. Applies complex queries; very fast; limited data volume.
- ROLAP: relational OLAP. Applies OLAP to relational data bases; large data volumes; slow
- HOLAP: hybrid OLAP. HOLPA can utilize both pre-calculated cubes (MOLAP) and relational data sources (ROLAP)
- WOLAP: Web-based OLAP
- SOLAP: Spatial OLPA
- DOLAP: Desktop OLAP
- OLAP Cloud

We can look at the information like a cube and we look for one slice of the cube for a product on a particular datawarehouse and then we can start comparing slices.

Ex: How much Chocolate did we sell in the South East in May 2016.

Did we sell more on that day or on this product?



## 6. Log formats

A log can be defined as a collection of events recorded during runtime of an information system.

Lately, event logs have become prominent as a basis for process analysis and mining. They store process data in an event-based way, i.e., for each observed execution of a process activity, at least one event is written to the corresponding process log.

It is crucial that the log contains information on the event order. Either it is based on time stamps connected with the events or the assumption holds that the order of the events within the log reflects the order in which they occurred during process execution.

Two process-oriented log formats

- Mining XML
- Extensible Event Stream (XES)

## 7. From Transactional Data Towards Analytical Data

As it is not constructive to analyse each portion of extracted data separately, the extracted data is to be cleaned and integrated. But first we have to decide about the **integration format**. The choice of the analytical format depends on the analysis questions and the key performance indicators. The choice of the integration format depends on the results of the data extraction step.

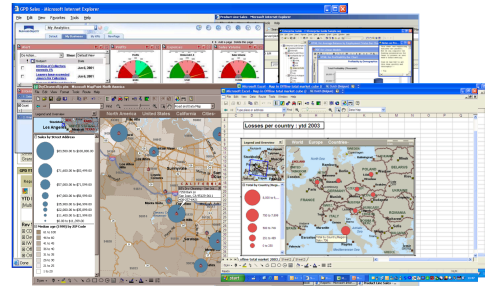
		Structured Data Formats			Unstructured Data
		Flat (e.g., relational, CSV)	Hierarchical (e.g., XML)	Hybrid (e.g., XES)	Text
Table Formats	Flat	Contains / generates (mapping)	Generates (mapping)	Contains	Mining and generation
	Multidimensional	Generates (mapping & aggregation)		Generates (mapping & aggregation)	
Log		Generates (mapping & transformation)	Generates (mapping & transformation)	Contains or generates (transformation)	

The cells describe how integration and analysis formats can be transferred to other integration and analysis formats (from top to bottom)

- **Aggregation:** Sum, AVG. Aggregation refers to defining different abstraction levels within the schema and aggregating the data along these levels.
- **Mapping:** dimensionality might be changed; describes a set of attribute correspondences between two schemata based on which one schema can be mapped onto the other.
- **Transformation:** transformation of one schema into another to obtain a desired target schema (format).

## 8. Reporting & Analysis

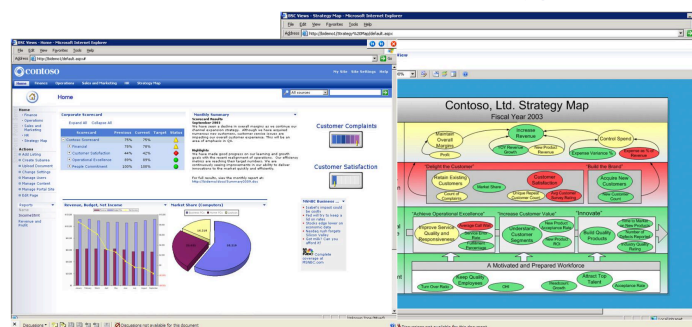
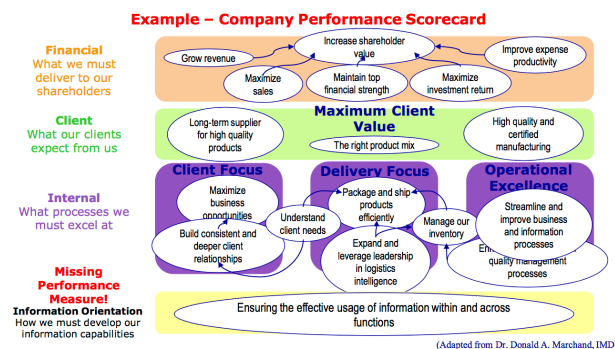
These are examples of reporting and analysis tools. There exist various techniques like cubes, visualisations, pivot tables, web access, Maps, Drill-down, etc.



**Enterprise reporting** is a popular business intelligence (BI) discipline that extends reporting and analysis capabilities beyond the scope of IT staff, business analysts, and power users. With enterprise reporting, anyone who impacts a business - executives, managers, analysts, and frontline workers - have immediate access to the vital information they need to most productively perform their jobs.

Briefly said, enterprise reporting are analytic applications that offer ready-made report templates for industry-specific metrics and thresholds for alerts (Mission critical)

### 1) Balanced Scorecard



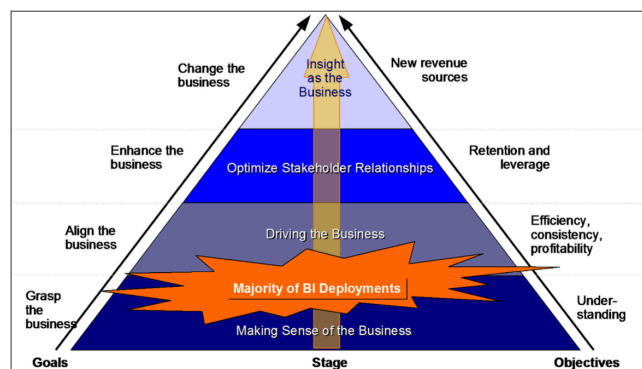
- 2) Digital Dashboard
- 3) Corporate Performance Management
  - a. CRM Analytics
  - b. Financial analysis
  - c. Supply Chain Intelligence (SCI)
- 4) Enterprise analytics
- 5) Business Activity Monitoring (BAM)

Business Process Tree	Last Reading	Time	Status	Details
<ul style="list-style-type: none"> <li>Pick to Order Direct <ul style="list-style-type: none"> <li>Order Capture</li> <li>Order Validation</li> <li>Order Import to ERP <ul style="list-style-type: none"> <li>Volume <ul style="list-style-type: none"> <li>Volume by Channel - Per Period</li> <li>Average Cycle Time - Per Period</li> </ul> </li> </ul> </li> <li>Number of Stuck Line Items <ul style="list-style-type: none"> <li>Cumulative</li> <li>Per Period</li> </ul> </li> <li>Number of Stuck Line Items by Channel <ul style="list-style-type: none"> <li>Call Center</li> <li>eBay B2B</li> <li>eBay Consumer</li> </ul> </li> </ul> </li> </ul>				
	725	10/01/03 8:00 AM		
	0:1:45	10/01/03 8:00 AM		
	44	10/01/03 8:00 AM		
	24	10/01/03 8:00 AM		
	0	10/01/03 8:00 AM		
	4	10/01/03 8:00 AM		
	0	10/01/03 8:00 AM		

6) Also: IT operations: IT's own scorecards

## Discovery

- **Verification driven analysis:** Analyst proposes possible patterns, verification by the analyst
  - Query and basic reporting
  - OLAP
  - Automation of Enterprise Reporting
  - Statistical techniques
- **Discovery driven data mining:** Automated search for patterns, verification partly automatic
  - Classification
  - Segmentation
  - Associations
  - Sequence analysis

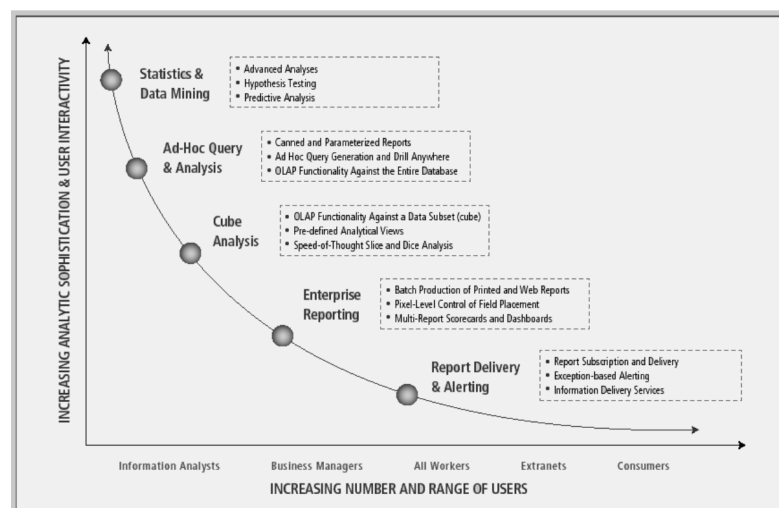
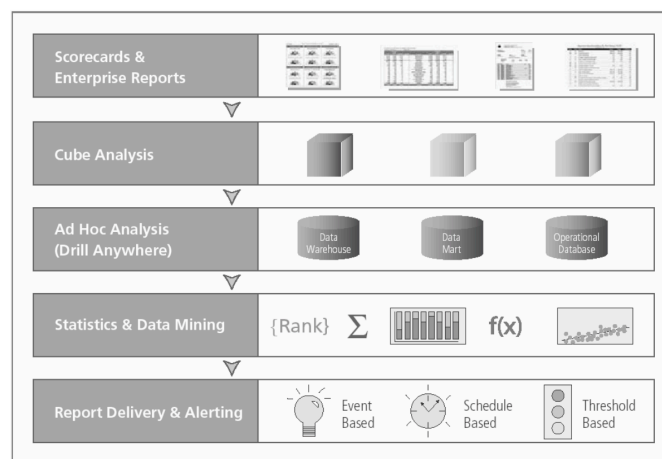


⇒ We are moving from a tactical scope to a strategic impact.

## 9. Applications

- 1) **Enterprise Reporting** – Broadly deployed report formats for operational reporting and scorecards/dashboards targeted at information consumers and executives.

- 2) **Cube Analysis** – OLAP slice-and-dice analysis of limited data sets, targeted at managers and others who need a safe and simple environment for basic data exploration within a limited range of data.
- 3) **Ad Hoc Query and Analysis** – Full investigative query into all data, as well as automated slice-and-dice OLAP analysis of the entire database – down to the transaction level of detail if necessary. Targeted at information explorers and power users.
- 4) **Statistical Analysis and Data Mining** – Full mathematical, financial, and statistical treatment of data for purposes of correlation analysis, trend analysis, financial analysis and projections. Targeted at the professional information analysts.
- 5) **Alerting and Report Delivery** – Proactive report delivery and alerting to very large populations based on schedules or event triggers in the database. Targeted at very large user populations of information consumers, both internal and external to the enterprise.

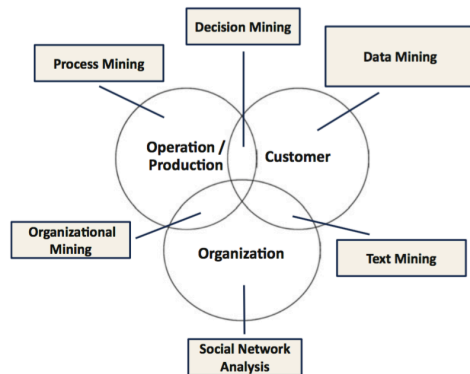




## CHAPTER 4: KNOWLEDGE DISCOVERY IN DATA (KDD)

### 1. What is KDD?

#### 1.1 Overview



This figure shows an overview of the different types of mining, in connection with the different BI perspectives. This includes mining algorithms that frequently occur in connection with overlapping BI perspectives.

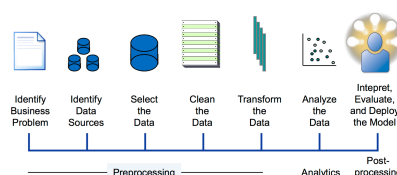
The choice of the algorithms depends on the view and the envisaged analytical goal.

**KDD** is the non-trivial process of identifying **valid**, **novel**, potentially **useful** and ultimately **understandable** patterns in data.

It is an iterative and multi-disciplinary process.

- **Valid:** Discovered patterns should be true on new data with some degree of certainty.
- **Novel:** Patterns must be novel (should not be previously known)
- **Useful:** Actionable; patterns should potentially lead to some useful actions.
- **Understandable:** The process should lead to human insight. Patterns must be made understandable in order to facilitate a better understanding of the underlying data.

KDD is the detection of non-trivial, implicit, previously unknown and possible useful knowledge from data.



As previously mentioned, KDD is an interactive and iterative process consisting of:

- Data selection, cleaning and transformation (pre-processing)
- Data mining
- Evaluation and interpretation of the obtained knowledge (post-processing)
- Integration with existing knowledge
- Making the knowledge accessible

The additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, are essential to ensure that useful knowledge is derived from the data.

Interesting application areas are:

- Direct Marketing
- Fraud detection
- Market basket analysis
- Stock picking
- Cross selling
- Web mining

## 1.2 Steps in the KDD process

- 1) Problem Analysis: analysing a business problem to assess if it is suitable for tackling using data mining.
- 2) Data preparation: data collection, aggregation, table joins, deriving new fields, data pre-processing, etc.
- 3) Data exploration: visualisation driven exploration to give the user a good feel for the data and to reveal any errors in the data preparation/Extraction.
- 4) Data mining: decide upon technique and run
- 5) Interpretation and validation of the generated output
- 6) Pattern deployment: use discovered knowledge in decision support systems, to produce reports/guidelines, or to filter data for further processing.

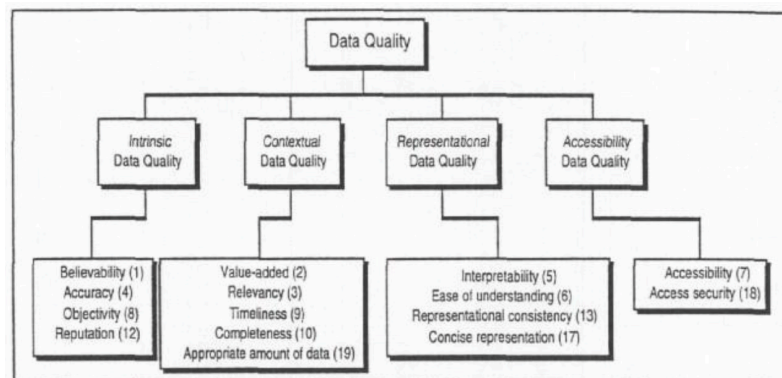
SEMMA (SAS institute)

- Sample (Training, Validation, Test)
- Explore (get an idea of the data at hand)
- Modify (select, transform)
- Model (create data mining model)
- Assess (validate model)

## 2. Data pre-processing

- How to sample? E.g. credit scoring: reject inference
- Types of attributes
  - Continuous: e.g. age, amount of loan, ...
  - Categorical: e.g. purpose of loan, marital status, ...
- Encoding of categorical variables
  - Dummies
  - Logarithmic encoding
  - Other
  - Beware: data explosion!
- Discretisation of attributes: E.g. by domain expert or algorithm of Fayyad and Irani
- Decide upon target attribute and its encoding  
E.g. credit scoring: definition of bad customer?
- Missing values: imputation procedures
- Outliers (extreme observations)  
E.g. Age of person is 3000 years • Keep or remove?
- Input selection

- Feature construction
- Data Integrity
  - Which Data Element with the following values is actual: 25, 67, 135?
  - What is the right value?
- Data Redundancy: How much data in your company is redundant?
- Timeliness of Data
  - Is your data up-to-date?
  - Is the worth of your data changing due to a lifecycle?
- Accessibility: Do you have access to all data you need?



### 3. Data mining tasks and techniques

#### 3.1 Predictive techniques

⇒ Supervised

##### 3.1.1 Classification

Goal: use attributes to classify subject into predefined class

Target class: discrete

Techniques & Models

- **Mathematical models**

$f(x) > 0.5 \Rightarrow \text{customer} = \text{good}$

$f(x) \leq 0.5 \Rightarrow \text{customer} = \text{bad}$

- **Linear**: Linear, logistic regression; linear discriminant analysis

Result: linear function of attributes

$f(x) = 0.125 \text{ income} + 0.305 \text{ age} - 0.02 \text{ gender} + 3.1 \text{ amount loan}$

- **Non-linear**

Artificial Neural Networks, Support Vector Machines, RVM, ...

Result: non-linear function of attributes

$f(x) = 0.201 \text{ income}^2 \text{ age}^3 - 0.55 \text{ age}^3 - 5.21 \text{ gender income} + 3.6 \text{ gender}^2 \text{ amount loan}^2$

- **Rule/tree based models**

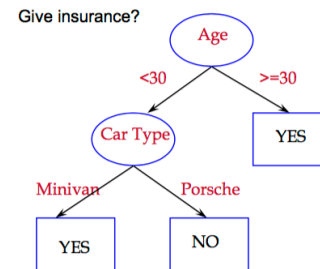
Decision Rules / Trees

C4.5, RIPPER, CN2, AntMiner+, ANN/SVM Rule extraction...

Result: set of rules or tree

```

if (Checking Account < 200 DM and Duration > 15 m and
    Credit History = no credits taken and Savings Account < 1000 DM)
then class = bad
else if (Purpose = new car/repairs/education others and
    Credit History = no credits taken/all credits paid back duly at this bank and
    Savings Account < 1000 DM)
then class = bad
else if (Checking Account < 0 DM and
    Purpose = furniture/durable appliances/business and
    Credit History = no credits taken/all credits paid back duly at this bank and
    Savings Account < 500 DM)
then class = bad
else if (Checking Account < 0 DM and Duration > 15 m and
    Credit History = delay in paying off in the past and
    Savings Account < 500 DM)
then class = bad
else class = good
  
```



Application: use income, age, ... to classify customer as credit-worthy or not

### Attention Points

- Misclassification costs: FP vs. FN
- Sample construction (skewed distribution). Ex: 99% good customers, 1% bad customers
- Performance of a classifier: how to define classification accuracy?
- How to judge performance differences?
- Black box versus white box classifiers
  - Explanatory power of the classifier
  - Ex: Neural networks vs. decision trees

### 3.1.2 Regression

Goal: predict future (continuous) values using information about the past.

Target: continuous variable

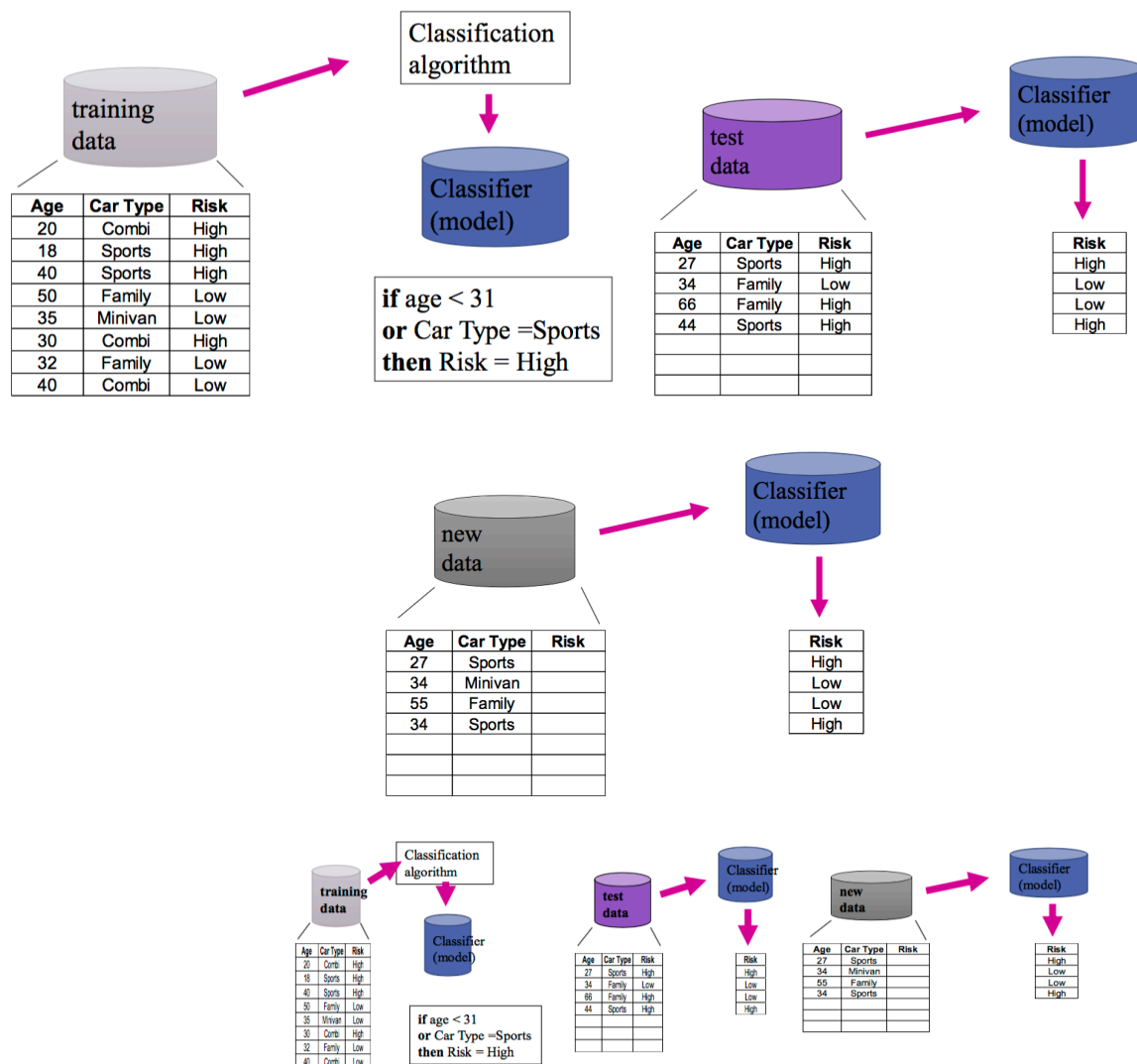
#### Techniques and models

- Mathematical models
  - Linear: linear logistic regression
  - Non-linear: artificial neural networks, SVM, ...
- Rule/tree based models
  - Regression trees (CART)

Application: predict stock prices, predict amount of sales, ...

#### Attention points:

- **Training set vs. Test set**

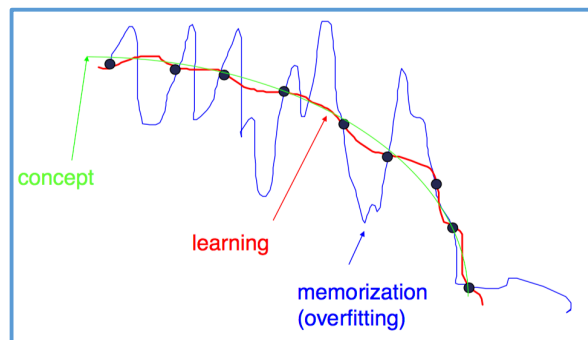


• **How to measure Generalisation behaviour ?**

- Split-sample (1/3 test)
- N-fold cross-validation

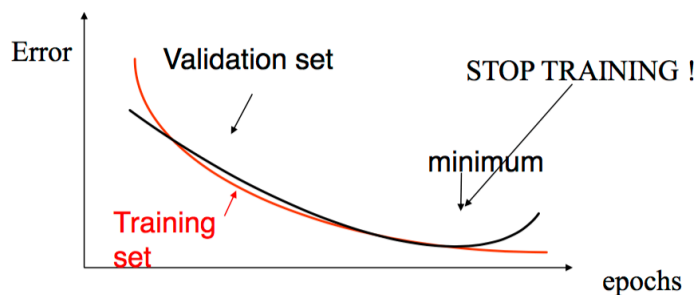
• **Learning vs. Overfitting**

- Successful learning: recognize data outside the training set, i.e. data in an independent test set
- Overfitting ("Memorization")
  - Each data set is characterized by noise (idiosyncrasies) due to incorrect entries, human errors, irrationalities, noisy sensors, ...
  - A model that is too complex e.g. neural network with too many neurons) may fit the noise, not just the signal, leading to overfitting.

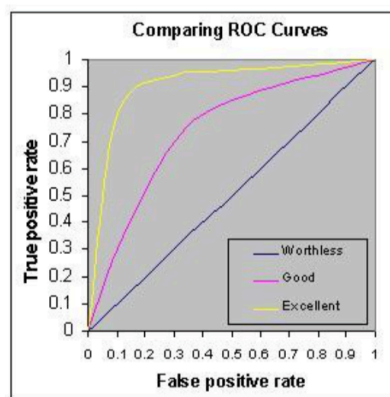


⇒ How to avoid overfitting?

Set aside a validation set. Use the training set to train parameters and validation set to stop training when overfitting occurs (“early stopping”).

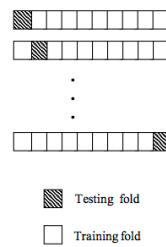


- Generalisation behaviour
  - How well does the trained model perform in predicting new, unseen (!) instances?
  - Decide upon performance measure
    - Classification: percentage correctly classified (PCC), sensitivity, specificity, area under ROC curve (AUROC)
    - Regression: Mean Absolute Deviation (MAD), Mean Squared Error (MSE), ...



- **Split-sample method:** set aside a set (typically 1/3th of the data) which is NOT used during training. Estimate generalisation behaviour of trained network on test set.
- **Single sample method:** generalisation behaviour =  $f(\text{training error, model complexity})$   
 Ex: Akaike Information Criterion, Bayesian Information Criterion, ...  
 Model complexity can be measured by the number of trained parameters. This method is based on statistical learning theory.

- **N-Fold Cross-Validation:** this method is intended for small data sets. Basically, you split the data in N-folds, typically 10. You then train on N-1 folds and test on remaining fold. Repeat this N-times and compute the mean of the performance measure.



### Assessing the Performance

- Measuring generalization behaviour
  - Training set for learning
  - Validation set for model selection
    - Decision trees: subtree when pruning
    - NN: number of hidden layers, number of neurons
    - SVM: parameter settings
  - Test set for independent evaluation
- Performance measures
  - **Accuracy:** PCC
  - **Comprehensibility.** Important aspect in domains as credit scoring and medical diagnostic  
 Equal Credit Opportunity Act : vague reasons for denial are illegal  
 Non-linear < Linear < Rule/Tree-based  
Preferably: Number of rules, Number of terms per rule low  
Solutions: Linear, Rule/Tree based models; ANN / SVM rule extraction
  - **Justifiability:** In line with existing knowledge  
 Linear classifiers: expected sign  
Ex:  $PD = 0.013 \times \text{age} + 1.23 \times \text{income} + \dots + 2.3 \times \text{amount}$   
 If recalling name street = yes and years education > 5 then patient = normal  
 Else if recalling name street = no and age > 80 then patient = normal  
 Else patient = dementia diagnosed  
Solutions:  
 Emitting terms:  $PD = 0.013 \times \text{age} + \dots + 2.3 \times \text{amount}$   
 If recalling name street = yes and years education > 5 then patient = normal  
 Else patient = dementia diagnosed  
 Change model

### 3.2 Descriptive techniques

⇒ Unsupervised

#### 3.2.1 Association rules

Goal: Detect frequently occurring patterns between items given a minimal level of support and confidence.

Techniques: deriving association rules; Apriori Algorithm

Application: Buying products together

Ex: if a customer buys spaghetti, then the customer buys red wine in 70% of the cases.

Notation:

- D: database of transactions  $tp$  (tuples). Each transaction  $tp$  consists of a transaction ID and a set of items  $\{i_1, i_2, \dots, i_n\}$  selected from all possible items  $I$   
An association rule is an implication of the form:  $X \rightarrow Y$  where  $X \subseteq I, Y \subseteq I$  and  $X \cap Y = \emptyset$
- Support of an itemset is the percentage of total transactions in the database that comprise the itemset. The rule  $X \Rightarrow Y$  has support  $s$  if  $s\%$  of the transactions in  $D$  contain  $X \cup Y$

$$\text{Sup}(X \Rightarrow Y) = \frac{\text{number transactions supporting } X \cup Y}{\text{total number of transactions}} = p(X \cup Y)$$

The rule  $X \Rightarrow Y$  has Confidence  $c$  if  $c\%$  of the transactions in  $D$  that contain  $X$  also contain  $Y$ .

$$\text{Conf}(X \Rightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)} = \frac{p(X \cup Y)}{p(X)} = p(Y|X)$$

The **Apriori algorithm** can be used to mine association rules. This algorithm is a two-step process

1. Identification of all (large) itemsets having support above minsup, i.e. « frequent » itemsets; (Apriori-Algorithm)
2. Discovery of all derived association rules having confidence above minconf;

⇒ **Minsup** and **minconf** need to be specified in advance!

Basic notion

- Every subset of a large itemset must be a large itemset (Apriori property)



- So candidate itemsets having  $k$  items can be found by joining large itemsets having  $(k-1)$  items and deleting those sets that contain any subset that is not large.
- This results in a much small number of candidate items

```

1)  $L_1 = \{\text{Large 1-itemsets}\};$ 
2) for ( $k=2; L_{k-1} \neq 0; k++$ ) do begin
3)    $C_k = \text{apriori-gen}(L_{k-1});$  // New candidates
4)   forall transaction  $t \in D$  do begin
5)      $C_t = \text{subset}(C_k, t);$  //Candidate contained in t
6)     forall candidates  $c \in C_t$  do
7)        $c.\text{count}++;$ 
8)   end
9)    $L_k = \{c \in C_k \mid c.\text{count} \geq \text{min sup}\}$ 
10) end
11)  $\text{Answer} = \bigcup_k L_k$ 

```

## Join Step

```

insert into  $C_k$ 
select  $p.\text{item}_1, p.\text{item}_2, \dots, p.\text{item}_{k-1}, q.\text{item}_{k-1}$ 
from  $L_{k-1} p, L_{k-1} q$ 
where  $p.\text{item}_1 = q.\text{item}_1, p.\text{item}_2 = q.\text{item}_2, \dots, p.\text{item}_{k-2} = q.\text{item}_{k-2},$ 
 $p.\text{item}_{k-1} < q.\text{item}_{k-1};$ 

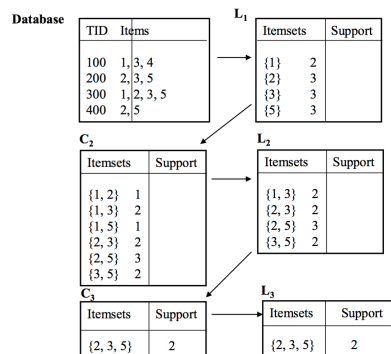
```

## Pruning Step

```

forall itemset  $c \in C_k$  do
  forall  $(k-1)$ -subsets  $s$  of  $c$  do
    if ( $s \notin L_{k-1}$ ) then
      delete  $c$  from  $C_k$ 

```



Minsup=50%

{1,3} and {2,3} give {1,2,3}  
but since {1,2} is not  
frequent we can prune it !

Result =  $\bigcup_k L_k$   
 $= \{ \{1,3\}, \{2,3\}, \{2,5\}, \{3,5\}, \{2,3,5\} \}$

Once the frequent itemsets have been obtained, the association rules can be generated as follows:

- For each frequent itemset  $I$ , generate all nonempty subsets of  $I$
- For every nonempty subset  $s$  of  $I$ , output the rule  $s \Rightarrow I - s$  if the confidence  $>$  minconf.

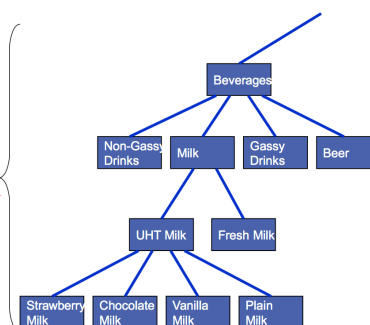
→ Multilevel association rules: mine association rules at different concept levels.

Mine association rules at  
different concept levels !

e.g. Chocolate and Milk  $\Rightarrow$  Beer

Product  
Taxonomy

Srikant R. & Agrawal R., "Mining  
Generalized  
Association Rules", In *Proc. 1995 Int. Conf.  
Very Large Data Bases*, Zurich, 1995.



### 3.2.2 Sequences

Detect **temporal patterns** between items.

Ex:

- Customer buys product X, then product Y, then product Z, ...
- 60% of clients who placed an online order in company/products/product1.html, also placed an online order in /company1/products/products4 within 15 days.

Technique: modified Apriori

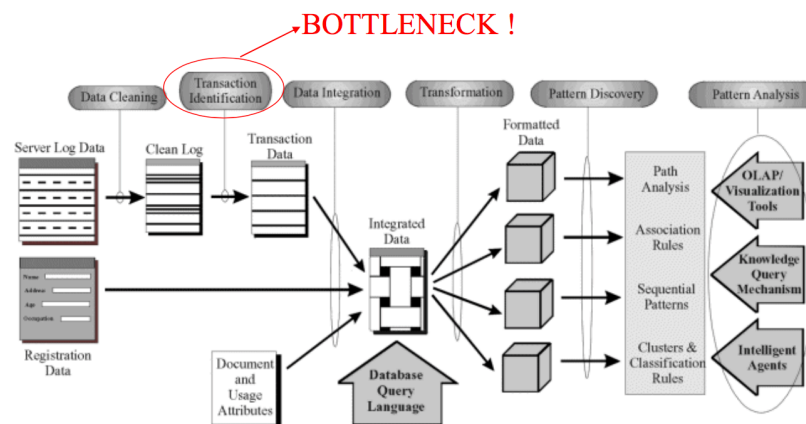
### 3.2.3 Web Mining

Web mining is the use of data mining techniques to automatically discover and extract information from Web documents and services.

- Mining Web log records to discover user access patterns of Web pages
- Improve web site design
- Identify potential prime advertisement locations
- Target customers for electronic commerce

Ex:

- 40% of clients who accessed the Web page with URL /company/products/product1.html, also accessed /company/products/product2.html;
- 30% of clients who accessed /company/announcements/special-offer.html, placed an online order in /company/products/product1.



### 3.2.4 Clustering

Goal: Identification of clusters derived from data (Unsupervised)

Method: Divide data into clusters such that

- Maximal similarity between items within cluster
- Maximal dissimilarity between items of different clusters
- Select features describing cluster

Techniques:

- Neural techniques

- Traditional statistical algorithms

#### Applications:

- Distinguishing market segments, customer profiles
- Often pre-processing to other techniques

## 4. Post-Processing

- Verify and validate patterns
  - **Verify**: did we build the system right?
  - **Validate**: did we build the right system?
- Visualise knowledge
  - OLAP, Gains chart, ROC curve, decision table, ...
- Sensitivity analysis
- Contrast mined knowledge with existing domain knowledge
- Build intelligent system

⇒ The Role of **Occam's razor** in data mining

Theory: Occam's razor is a principle from philosophy. Suppose there exist two explanations for an occurrence. In this case, the simpler one is usually better. Another way of saying it is that the more assumptions you have to make, the more unlikely an explanation is.

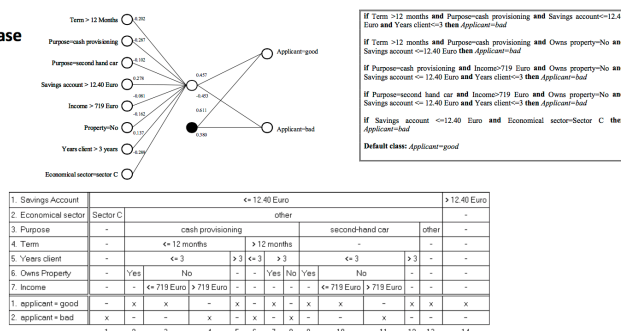
- Prefer the simplest model!
- Accuracy versus complexity trade-off
- Interpretability is important for validation!
- Input Selection
- Avoid using black box techniques as final data mining models

⇒ *“Perfection is achieved, not when there is nothing more to add but when there is nothing left to take away” Antoine de Saint Exupéry*

## 5. Applications

### 5.1 Credit Scoring

#### • Credit scoring case



## 5.2 Fraud detection

- Identify wrong actions (known fraud): similarity o known cases
- Identify actions by the wrong people
- Identify suspect actions: legal, but probably not right
- Identify wrong actions (unknown fraud)
  - Prevent wrong actions
  - Detect exceptions

How can automatic discovery help?

- Detecting and preventing fraud, waste and abuse
  - Learn from the past: high quality, evidence based decisions
  - Predict: prevent future instances
  - React to changing circumstances: models kept current, from latest data
- Payment error prevention
- Billing and payment fraud
- Audit selection
- Tax and insurance fraud

Applications: health care, credit card service, telecom

In KDD

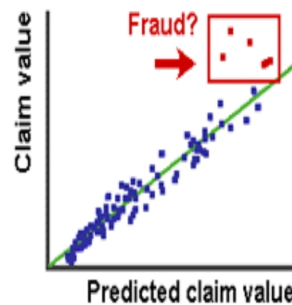
- Classification?
  - Classify into fraudulent and non-fraudulent behaviour
  - What do we need to do this?
- Prediction and Outlier Detection (anomalies/irregularities)
  - Assume non-fraudulent behaviour is normal
  - Find the exceptions
  - Can we rate or score cases on their degree of anomaly?
- Matching known fraud/non-compliance
  - Which new cases are similar to known cases?
  - How can we define similarity?
  - How can we rate or score similarity?
- Forecasting what may happen in the future
- Classifying people or things into groups by recognizing patterns
- Clustering people or things into groups based on their attributes
- Associating what events are likely to occur together
- Sequencing what events are likely to lead to later events

Techniques for identifying fraud

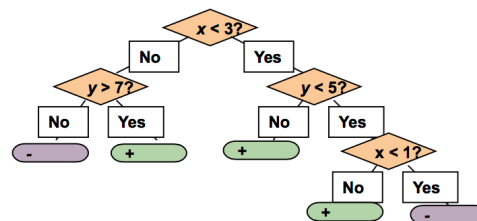
Predict and classify <i>Known output/Fraud</i>	Group and Associate <i>Unknown output/Fraud</i>
---	--

<ul style="list-style-type: none"> <li>- Regression algorithms: NN, CART, regression (predict numeric outcome)</li> <li>- Classification algorithms: C5.0, logistic regression (predict symbolic outcome)</li> </ul>	<ul style="list-style-type: none"> <li>- Clustering/grouping algorithms: k-means, Kohonen, 2step, factor analysis</li> <li>- Association algorithms: Apriori, sequence</li> </ul>
--	---

- 1) **Prediction:** predict the expected value for a claim, compare that with the actual value of the claim. Those cases that fall far outside the expected range should be evaluated more closely.



- 2) **Classification:** build a profile of the characteristics of fraudulent behaviour. Pull out the cases that meet the historical characteristics of fraud.



- 3) **Clustering:** Group behaviour using a clustering algorithm. Find 'natural' grouping of instances given unlabelled data. Then identify outliers and investigate.

**How?**

- Maximize intra cluster similarity
- Minimize inter cluster similarity
- Select features describing cluster

- 4) **Profiling:** cluster users, then develop profiles for clusters. Give profile of individual behaviour.

Existing users: do they match profile for their cluster?

New user: Do they match any profile?

## 6. Issues and conclusions

To conclude ...

- KDD allows to fully exploit available data
- KDD is a very technical exercise requiring specialised expertise
- Continuously in motion

- More applied research is needed
- Role of the domain expert remains important
- Hard to quantify ROI and KDD

Final issues:

- **Garbage in – garbage out**
- **Privacy**
- Role of the **domain expert** remains important
  - Pre-processing: how to clean up data, Which variables, transformations? Where to get data, ... ?
  - Data mining: which technique to use?
  - Post-processing: model accurate, comprehensible, justifiable (enough)? How to implement?

## CHAPTER 5: DECISION TREE INDUCTION

### 1. Introduction: learning tasks

Decision trees are part of the classification models. As a reminder, for classification models, the response variable has only a finite number of values  $Y$ . We want to learn a rule how the class membership of an observation can be predicted using the explanatory variables  $X$ .

Decision trees are used for the classification of cases or the prediction of outcomes.

- Induction of decision trees based on examples. The algorithm has to learn, based on what it learned it can build a tree.
- Algorithms: ID3, C4.5, CART
- The prediction of the class of an object on the bases of some attributes.

Ex:

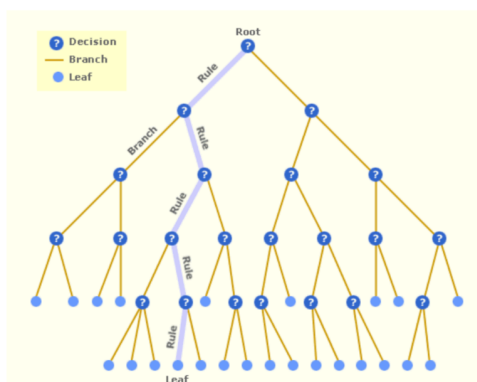
- Good or bad credit for loan applicants using
  - Income
  - Age
  - ...
- Spam/no spam for e-mail messages, using
  - % of words matching a given word
  - use of capital letter
  - ...

There are many different ways to build a classification tree.

- Logistic regression
- NN
- SVM
- ...

It can also be used for

- Regression: regression tree
- Clustering



Tree classifiers all do recursive partitioning of the data. This means that we build small trees, step by step, splitting on some attributes.

Decisions are taken by going down the tree, from top to bottom.

Purpose: build the best tree for a given set of examples.

By asking a sequence of questions that are answered with yes or no, the candidate tries to limit the number of possible terms until he/she can make a guess with high confidentiality.

We use a binary tree for modelling purposes. All cases of the training data belong to the root node. In each node of the tree, the data belonging to that node is split into subsets according to the values of one input variable.

Algorithm: CART = Classification And Regression Trees

## 2. Method and terminology

### Target

- Discrete variable: classification tree
- Continuous variable: regression tree

### Recursive partitioning

- Find the most important variable
- Split into subsamples (as homogeneous as possible with regard to the outcome)
- Repeat procedure recursively on each subsample
- Stop when the prediction is sufficiently strong

The result of this process is a **set of rules**.

⇒ Apply rules to test data (containing target values) to estimate error/compare with other techniques

Score new data

### 3 questions

#### 1) How to split a node = **splitting rule**

Splitting rules: A strategy for growing the tree defining in each node which variable should be used for splitting together with the threshold for the split.

- Calculate all possible splits
- Calculate the goodness of the split
- Choose the best split

Look at all possible splits for a variable and this for all the variables. Desirable splits are the ones for which the distributions of the outcome in the child nodes are more homogenous (purer) than the root node.

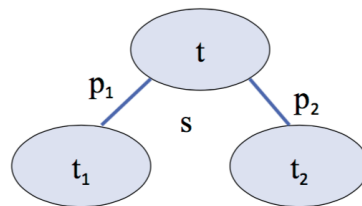
Splitting is based on measures for the impurity of a node. A node has impurity 0 if only input data of one class belong to the node. In the case of a metric variable, the split is defined by a decision rule in the form  $X < tr$  OR  $X \geq tr$ ; in the case of nominal variables, we decide according to the rule  $X = a$  OR  $X$  different from  $a$ .

For the split, the variable is used, which minimizes the impurity in the child nodes.

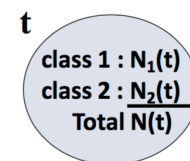


### Goodness of the split

- Consider candidate split  $s$  of node  $t$
- $P_i$  is the proportion of cases sent to  $t_i$
- Denote  $I(t)$  as the impurity of node  $t$ : then the goodness of the split is weighted mean decrease in impurity  $\Delta I(s,t) = I(t) - p_1 I(t_1) - p_2 I(t_2)$
- Tree impurity  $I(T) = \sum_t I(t)$



- Impurity  $I(t)$  of node  $t$**   
= function of the nodes conditional probabilities
- 2 popular measures :
  - Entropy measure (Shannon)**
    - $I(t) = -\sum_j p(j|t) \log[p(j|t)]$
    - If 2 target classes  
 $I(t) = -p(1|t) \log(p(1|t)) - p(2|t) \log(p(2|t))$
  - Gini index of diversity**
    - $I(t) = \sum_{i \neq j} p(i|t) p(j|t)$
    - If 2 target classes  
 $I(t) = 2 p(1|t) p(2|t)$



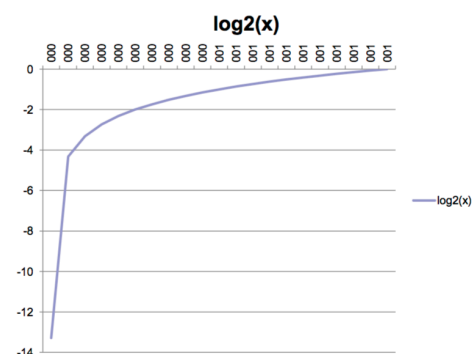
$$p(1|t) = N_1(t) / N(t)$$

conditional probability

attribute	value	# yes	# no	entropy
outlook	overcast	5	3	0.72
	clear	4	5	0.92
	rain	3	2	0.92
temperature	hot	2	2	1.00
	mod	4	2	0.92
humidity	high	3	1	0.92
	normal	2	4	0.92
wind	strong	0	1	0.00
	weak	5	3	0.92

largest when  $N_1(t) = N_2(t)$   
smallest when  $N_1(t)=0$

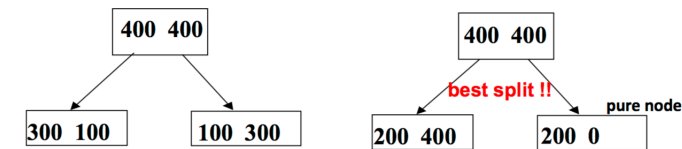
x	$\log_2(x)$
0	$-\infty$
0.1	-3.32193
0.2	-2.32193
0.3	-1.73697
0.4	-1.32193
0.5	-1
0.6	-0.73697
0.7	-0.51457
0.8	-0.32193
0.9	-0.152
1	0



### Impurity $I(t)$ of node $t$

- At maximum when observations are distributed evenly over all classes
- At minimum when all observations belong to a single class
- A symmetric function of  $p(1|t), p(2|t), \dots, p(J|t)$ .

The impurity of a node is the amount of confusion: whether it is yes or no does not matter, it is the amount of yes and no that matters.



t = impurity of top node  
l = impurity of left node  
r = impurity of right node

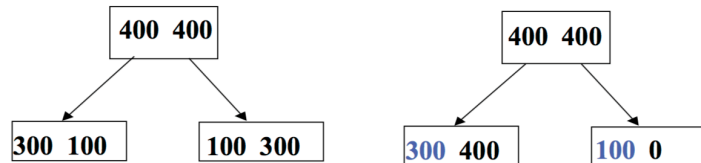
$$\text{Gini: } I(t) = 2 p(1|t) p(2|t)$$

$$\begin{aligned} t &= 2 \times (1/2) \times (1/2) = 0,5 \\ l &= 2 \times (3/4) \times (1/4) = 0,375 \\ r &= 2 \times (1/4) \times (3/4) = 0,375 \end{aligned}$$

$$\begin{aligned} \Delta I(t) &= t - (400/800) \times l - (400/800) \times r \\ &= 0,5 - (1/2) \times 0,375 - (1/2) \times 0,375 \\ &= 0,125 \end{aligned}$$

$$\begin{aligned} t &= 2 \times (1/2) \times (1/2) = 0,5 \\ l &= 2 \times (1/3) \times (2/3) = 0,44 \\ r &= 2 \times (1) \times (0) = 0 \end{aligned}$$

$$\begin{aligned} \Delta I(t) &= t - (600/800) \times l - (200/800) \times r \\ &= 0,5 - (3/4) \times 0,44 - 0 \\ &= 0,166 \text{ (larger reduction)} \end{aligned}$$



$$\begin{aligned} t &= 2 \times (1/2) \times (1/2) = 0,5 \\ l &= 2 \times (3/7) \times (4/7) = 0,48 \\ r &= 2 \times (1) \times (0) = 0 \end{aligned}$$

$$\begin{aligned} \Delta I(t) &= \\ &= 0,125 \end{aligned}$$

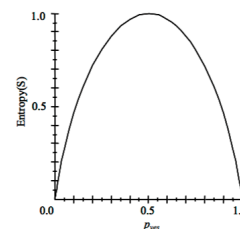
$$\begin{aligned} \Delta I(t) &= t - (700/800) \times l - (100/800) \times r \\ &= 0,5 - (7/8) \times 0,48 - 0 \\ &= 0,07 \end{aligned}$$

- Entropy: measure for impurity

$$E(S) = -p_{yes} \log_2 p_{yes} - p_{no} \log_2 p_{no}$$

$$E(S) = -\sum_{k=1}^k p_k \log_2 p_k$$

attribute	value	# yes	# no	entropy
outlook	sunny	2	3	0.97
	overcast	4	0	0.00
	rain	3	2	0.97
temperature	hot	2	2	1.00
	mild	4	2	0.92
	cold	3	1	0.91
humidity	high	3	4	0.99
	normal	6	1	0.59
wind	weak	6	2	0.91
	strong	3	3	1.00



- Information Gain

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

attribute	remaining entropy	information gain
outlook	0.69	0.25
temperature	0.91	0.03
humidity	0.79	0.15
wind	0.89	0.05

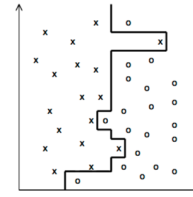
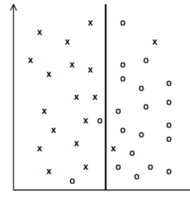
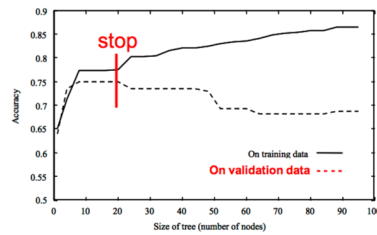
Decision tree induction

- ID3
  - Splitting rule = information gain
  - Drawback: favours attributes with many values
- C4.5
  - Splitting rule = gain ratio

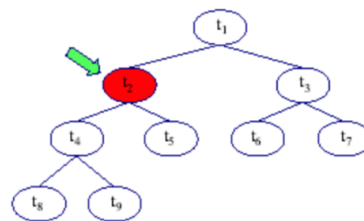
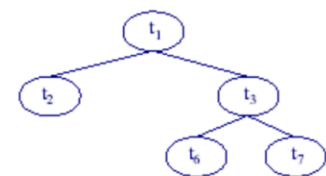
$$\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInformation}(S, A)}$$

$$\text{SplitInformation}(S, A) = -\sum_{i=1}^c \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

## 2) When do we declare a node terminal = **stopping rule**



- Early stopping
  - Pruning: grow a very large tree and merge back nodes.
- Pruning rule:** A strategy for pruning the tree in order to avoid overfitting to the training data.
- Done in C4.5
  - Ex: Tree T

Pruning Tree T in node  $t_2$ T after pruning in  $t_2$ 

What is the right size for a tree?

Idea: select the pruned subtree that has the lowest error (= misclassification rate) on the sample.

Therefore, split the sample in

- Training sample: use this to grow the tree
  - Validation sample: use this to evaluate the subtrees
- Gives us a second preclassified data set drawn from same population but with different records winner = subtree that classifies the validation data the best.
- Ex: 70% training data – 30% validation data

## 3) Inferences on terminal nodes = **assignment rule**

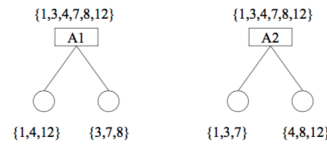
Once we have grown a tree, even if it is not 100% accurate, we have to decide what to put at the end. What class to assign to each leaf?

- Largest distribution
- Taking into account misclassification cost

Regression and clustering trees

- Regression tree:
  - Splitting rule: variance of values

- Assignment: mean value of leaf
- Clustering tree:
  - Splitting rule: mean squared distances of samples
  - Assignment: none – unsupervised learning



### 3. Advantages and disadvantages of decision trees

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>- Popular (top-3 most used data mining techniques after logistic regression)</li> <li>- Ease of interpretation</li> <li>- Results are invariant with respect to monotone transformations</li> <li>- Ability to handle both continuous and categorical variables</li> <li>- Extremely robust for the effect of outliers. It does not matter too much if we have a few lines that are totally wrong.</li> <li>- Performs well on large datasets.</li> </ul>	<ul style="list-style-type: none"> <li>- Classification trees may be unstable: although these splits may almost have the same goodness of split they can be very dissimilar</li> <li>- Classification trees may give a too complex tree, which is difficult to comprehend and does not generalize well (overfitting). Importance of pruning</li> <li>- Computationally expensive to train</li> <li>- Pruning algorithms can also be expensive</li> <li>- Some concepts still hard to learn for decision trees: importance of feature engineering still valid</li> <li>- Information gain measure is biased towards attributes with more levels</li> <li>- Relatively robust to class imbalance, though still important to watch out for (Ex: 5 fraudsters vss. 1000 non-fraudsters)</li> </ul>

#### 1) Disadvantage: Overfitting

Pruning. Test on test set. Conditional inference trees: avoids the variable selection bias of selecting variables that have many possible splits or many missing values. This approach uses a significance test procedure in order to select variables instead of selecting the variable that maximizes an information measure. Statistical theory test for variable selection helps with overfitting.

#### 2) Disadvantage: Learning problem

Ex: linear relationships in CART; XOR-style interactions; Tree grows larger

#### 3) Disadvantage: Information gain bias. Information gain measure is biased towards attributes with more levels or attributes with many missing variables.

Conditional inference trees: avoids the variable selection bias of selecting variables that have many possible splits or many missing values. This approach uses a

significance test procedure in order to select variables instead of selecting the variable that maximizes an information measure.

Also: other measures exist as well (see Gini before)

Another common approach: force binary splits

## CHAPTER 6: KNOWLEDGE BASED SYSTEMS

### 1. KBS - An Overview

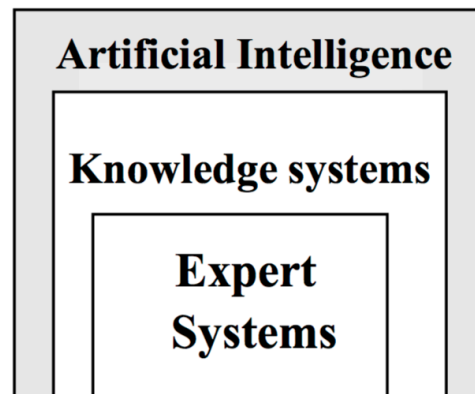
A knowledge-based system (KBS) is a computer program that reasons and uses a knowledge base to solve complex problems. The term is broad and is used to refer to many different kinds of systems; the one common theme that unites all knowledge based systems is an attempt to represent knowledge explicitly via tools such as ontologies and rules rather than implicitly via code the way a conventional computer program does.

**Knowledge based systems** are AI-systems that use a clearly defined body of knowledge to obtain their goals.

As a reminder, AI tries to make computers behave as if they were human beings: speech, movement, etc.

Knowledge based systems are a small form of AI.

- The knowledge about the problem domain is clearly separated from the procedural parts of such a system
- This architecture distinguishes knowledge based systems from classical systems where all the knowledge and procedures are tightly interconnected.



- **Knowledge based system: KBS**
  - Often used as a synonym for expert system. A knowledge based system is a more general kind of system.
  - It also uses a symbolic representation of knowledge and rules-of-thumb to exhibit a form of intelligent behaviour but it does not have to contain any real expertise.
- **Expert System (ES)**

An expert system is a special type of knowledge based system that contains a body of knowledge that normally belongs to an expert.

The system uses this knowledge in the same way and for the same purposes as real experts do.

An expert system is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution.

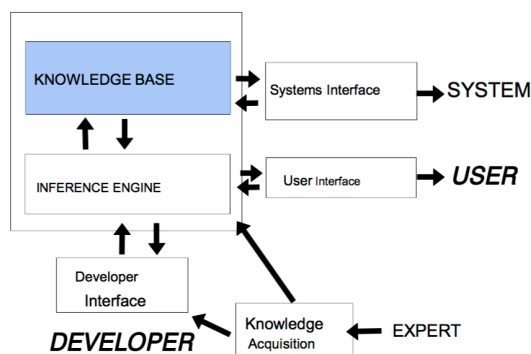
Why knowledge based systems (vs Human Expertise)?

- Knowledge is permanent
- Available at any time

- Consistent and reproducible decision making
- No constant rehearsal necessary
- Easy and cheap to transfer
- Easy to document (training)

>< BUT:

- Creativity, flexibility?
- Common sense knowledge?
- Broad focus?
- Responsibility?



1) Knowledge is separated from the inference and control

2) Different ways to represent knowledge

3) Heuristic search

- Blue Box: knowledge can be identified. Once you have knowledge in a set of rules, you can use it. We want to take away the building of a system and bring knowledge to it.
- User interface: way to communicate with users.

### Knowledge

- R1. IF DISTANCE > 5 then MEANS is "Drive"
- R2. IF DISTANCE > 1 and TIME < 15 then MEANS is "Drive"
- R3. IF DISTANCE > 1 and TIME > 15 then MEANS is "Walk"
- R4. IF MEANS is "Drive" and LOCATION is "Downtown" then ACTION is "take a cab"
- R5. IF MEANS is "Drive" and LOCATION is not "Downtown" then ACTION is "drive your car"
- R6. IF MEANS is "Walk" and WEATHER is "Bad" then ACTION is "take a coat and walk"
- R7. IF MEANS is "Walk" and WEATHER is "Good" then ACTION is "walk"

The knowledge could be a set of rules. The rules give us some interesting information.

Ex: we want the rule to determine the means of transportation. If distance > 5 then drive.

Rules are not IT related, which means that every business person could easily understand these rules.

Ex: Wine recommender system. The system will ask a number of questions. We first have some general questions. The next question are then more precise, the order depends on the answer of previous questions.

## 2. Reasoning

Run through the rules and try to find out which question to ask next. Ex: if flavour of meal is strong, advice red wine.

- **FACTS**

*Socrates is human*

- **RULES**

*X is human => X is mortal*

- **Object - Attribute - Value (O-A-V)**  
**Schema**  
**Attribute - Value (A-V) Pairs**

- **Premises**    **If**    .....    **and**  
                                 .....    **and**  
                                 .....    **and**  
**Conclusions**    **Then**    .....    [...]

- **INFERENCE**

*Socrates is human*

*X is human => X is mortal*

=> (**unification : substitute Socrates for X**)

*Socrates is mortal*

(**modus ponens**)

There are mainly 2 method used when using an inference engine: **forward and backward chaining**.

### 2.1 Forward chaining (used for planning)

The system considers all the rules and finds that only the conditions of Rule 1 are satisfied, given the known facts.

Fact 1 : DISTANCE = 7 (given)

Rule 1 is executed and Fact 3 is derived:

Fact 3 : MEANS = "Drive" (From Rule 1)  
All the rules are reconsidered and the system finds that only the conditions in Rule 4 are satisfied. Rule 4 is executed and a new fact is obtained.

Fact 4 : ACTION = "Take a cab" (From Rule 4)  
The system cannot deduce any other conclusions, so the reasoning stops and the conclusions are :

Fact 1: DISTANCE = 7  
Fact 2: LOCATION = "Downtown"  
Fact 3: MEANS = "Drive"  
Fact 4: ACTION = "Take a cab"

It starts with available data and uses inference rules to extract more data until a goal is reached. The engine searches a true "if" and then concludes "then" clauses resulting in additional information.

→ Start with data, reason until answer

take data, apply it to all the rules and try to do as much as you can with the data.

## Inference: Forward Chaining

- **Example**

R1: IF A and C THEN E  
R2: IF D and C THEN F  
R3: IF B and E THEN F  
R4: IF B THEN C  
R5: IF F THEN G

Given facts:  
A is true  
B is true  
What can be concluded?

- **Cycle through rules, looking for rules whose premise matches the working memory.**

	<u>Working memory</u>
	A, B
R4 fires: assert new fact C	A, B, C
R1 fires: assert new fact E	A, B, C, E
R3 fires: assert new fact F	A, B, C, E, F
R5 fires: assert new fact G	A, B, C, E, F, G

- **Concludes everything possible from available information/facts**



## Forward Chaining: Example

R1: IF the patient has a sore throat  
AND we suspect a bacterial infection  
THEN we believe the patient has strep throat

R2: IF the patient's temperature is > 100  
THEN the patient has fever

R3: IF the patient has been sick for over a month  
AND the patient has fever  
THEN we suspect a bacterial infection

Facts:  
Patients temperature=102  
Has been sick for 2 months  
Has a sore throat

Cycle 1: Consider R1, R2, R3  
R2 fires: assert patient has fever

Cycle 2: consider R1, R3  
R3 fires: assert bacterial infection

Cycle 3: R1 fires: assert strep throat

*Data driven reasoning: will fire all rules possible, can continue reasoning about irrelevant details.*

## 2.2 Backward chaining (used for diagnosis)

### Main Goal : Find value for ACTION.

The system looks for rules that have anything to say about ACTION. As it turns out, rules 4,5,6 and 7 give a value to ACTION. Using a criterion to solve the conflict between these rules (this will be discussed later), the system picks a rule. Suppose rule 7 is chosen.  
**Rule 7:** To be able to conclude about rule 7, the system needs values for MEANS and WEATHER. MEANS and WEATHER become the system's **subgoals**.

### Subgoal 1: Find value for MEANS.

The system looks for rules that have anything to say about MEANS. Rules 1, 2 and 3 find a value for MEANS. Again a criterion is used to resolve the conflict between these rules and Rule 3 is chosen.

**Rule 3:** To be able to conclude about Rule 3, the system needs values for DISTANCE and TIME. These become the new subgoals.

### Subgoal 1.1: Find value for DISTANCE

Since the value for DISTANCE is known to be 7, this subgoal is easily resolved

Starts with a list of goals and works backwards from the consequent (then) to the antecedent (if) to see if there is data available that will support any consequent. If the if clause is not known to be true, add it to the goals.

Easy to use if you know the goal.

## Inference: Backward chaining

- Attempts to prove a hypothesis (goal) by gathering supporting information

Example

R1: IF B and C THEN G      R4: IF E or F THEN C  
R2: IF A and G THEN I      R5: IF D and C THEN K  
R3: IF D and G THEN J

**Goal: I**

Goal I: need to trigger R2

Need both A and G

Subgoal A: need user input (ask user)

Subgoal G: Need to trigger R1

Need both B and C

Subgoal B: need user input (ask user)

Subgoal C: Need to trigger R4

Need E or F

Subgoal E: need user input (ask user)

## 3. Main knowledge representation forms

- Simple rules and facts (O-A-V)

OBJECT	ATTRIBUTE	VALUE
Car	Color Make # Cylinders Doors	Red Toyota 4 2

- **Structured rules**

- **Example**  
IF GMAT score  $\geq$  600  
THEN Admit student to MBA program  
ELSE do not admit student.
- **Example (conjunctive condition clauses)**  
IF sky is clear  
AND temperature is low  
THEN chance of frost is high
- **Example (disjunctive condition clauses)**  
IF age of car is new  
OR condition of car is good  
THEN car should start  
AND trip should be safe      OR in THEN-part?

### Inference techniques

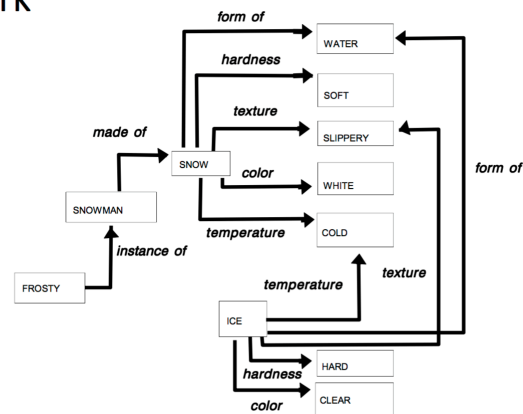
- **Deductive Reasoning** (Top-down logic): combines facts (axioms) with general knowledge in the form of implications to conclude new facts.  
Ex:  
Axiom: I am sleeping in class  
Implication: Sleeping in class  $\Rightarrow$  Rude shock in exams  
Conclusion: I will get a rude shock in the exams  
If A is true, and if  $A \Rightarrow B$  then B is true
- **Inductive reasoning** (Bottom-up logic): while the truth of conclusion of an inductive argument is probable, the one of deductive is certain.  
Generalizing from specific facts.  
Case 1: Game on 21<sup>st</sup> Sept. (Friday), it Rained, we lost  
Case 2: Game on 5<sup>th</sup> Nov. (Friday), it rained, we lost.  
...  
Induce general rule: if game is on a Friday AND it rains, then we lose.
- **Abductive reasoning:** Goes from an observation to a theory which accounts for the observation, to find simplest and most likely explanation. It is a deduction with plausible implications.  
If B is true, and  $A \Rightarrow B$ , then A is true.
- **Monotonic Reasoning:** facts remain static over period of problem-solving. Adding knowledge does not reduce the set of proposition that can be derived.
- **Non-monotonic reasoning:** "Having additional knowledge could allow less conclusions". Some conclusions can be invalid by adding knowledge.  
Ex: Truth Maintenance Systems used for non-monotonic reasoning.

- **Semantic networks**

**A semantic network**, or frame network, is a network that represents semantic relations between concepts. This is often used as a form of knowledge representation. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges, which represent semantic relations between concepts.

## Semantic Network

**Verbal Knowledge:** Frosty is a snowman. Snowmen are made of snow. Snow is a soft form of water. The texture of snow is slippery and its color is white. Snow is a cold substance. Ice is a cold substance, too. It is also a form of water with a slippery texture. Ice is hard and its color is clear.



- **Frames (inheritance)**

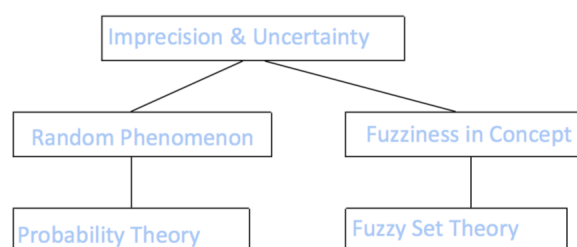
Frame systems attempt to reason about classes of objects by using **prototypical representations** of knowledge which hold good for the majority of cases, but which may need to be deformed in some way to capture the complexities of the real world. A frame is a description of an object that contains slots for all of the information associated with the objects. Slots, like attributes, may store values. Slots may also contain default values, pointers to other frames, sets of rules, or procedures by which values may be obtained. The inclusion of these additional features make frames different from object-attribute-value triplets.

Frame based systems are more powerful, but also more complex and more difficult to develop than simpler object-attribute-value/rule systems.

- **Objects**

- **Fuzzy Reasoning**

In daily life, people have to think and reason with imprecision & uncertainty in many cases;



Examples of fuzzy (linguistic) terms:

most, more-or-less, much more than, tall, good, very good, very tall, approximately, about, nearly, .....

Approximation

A' = " today is more or less sunny"  
B' = " sky is more or less blue"

illustration

Premise 1 (fact): x is A'

Premise 2 (rule): if x is A then y is B

---

Consequence: y is B'

(approximate reasoning or fuzzy reasoning!)

- **Logic**
  - Deduction and meta-rules
  - Declarative clauses

## CHAPTER 7: REGRESSION

⇒ Supervised; predictive

### 1. Model formulation and Terminology

Regression models are used for the prediction of the value of a response variable in dependence of a number of explanatory variables.

The first step in the analysis is the choice of an appropriate model class. We only consider two classes here, namely the linear regression and the logistic regression.

Linear regression models are used if we want to predict a metric response variable  $Y$ , for instance, a KPI, depending on a number of predictor variables  $X = (X_1, X_2, \dots, X_p)$ .

$$Y = f(x) + e$$

Where  $e$  is a random component describing the variation of the observed data. We assume a normal distribution with zero mean.

### 2. Linear Regression

⇒ Most prominent predictive model for finding the value of an output variable  $Y$  in dependence on the input variables  $X$ .

Given the input variables, the model class is defined by linear functions in the input variables, i.e.,  $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \varepsilon$ .

The interpretation of the models and the parameters is obvious in the case of quantitative input variables. The coefficients measure the effect per unit of the variable on the response. In the case of qualitative inputs (e.g. gender), the coefficients measure the effect of the categories represented by the dummy variable.

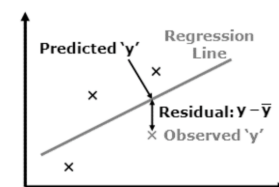
→ Select input variables, non-present variables' coefficients are represented in the model by 0.

Simple linear regression

Goal: : fit a line to data points

- $y = \beta_0 + \beta_1 x + \varepsilon$  with  $\varepsilon \sim N(0, \sigma)$
- The coefficients  $\beta_0, \beta_1$  are estimated using maximum likelihood estimation

$R^2$ : coefficient of determination, the proportion of variation in  $y$  explained by the model.



### Multiple Linear Regression

- When there are more than one independent variable
- $y = \beta_0 + \beta_1 x + \beta_2 x + \dots + \varepsilon$  with  $\varepsilon \sim N(0, \sigma)$
- The coefficients  $\beta_0, \beta_1, \dots$  are estimated using maximum likelihood estimation
- In general, smaller models are preferred
  - Use smallest subset of variables as possible whilst keeping  $R^2$  high
  - Stepwise introduction or removal of variables
  - Keep significant variables only

### Other validation checks

- Check residuals/fit of model
- Variables with too extreme coefficients
- Sign of the coefficients

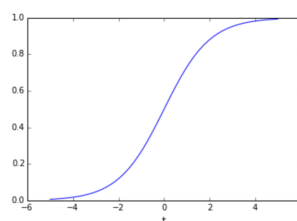
## 3. Logistic Regression

In statistics, **logistic regression**, or logit regression, or logit model is a regression model where the dependent variable (DV) is categorical. This article covers the case of a binary dependent variable—that is, where it can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. Cases where the dependent variable has more than two outcome categories may be analysed in multinomial logistic regression, or, if the multiple categories are ordered, in ordinal logistic regression.

For categorical variables, such as classification, this model is similar to linear regression. It uses the logistic function of the independent variables. The coefficients are usually estimated by the maximum likelihood method.

$$P(y = 1|x) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x + \beta_2 x + \dots))}$$

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Easy to interpret and understand</li> <li>• Statistical rigor</li> <li>• Relatively robust to noise</li> </ul>	<ul style="list-style-type: none"> <li>• Only two categories (otherwise use multinomial lr)</li> <li>• Sensitive to outliers</li> <li>• Categorical variables need to be converted to dummies</li> </ul>



## CHAPTER 8: CLUSTERING

⇒ Unsupervised; Descriptive

### 1. Unsupervised Data Mining

**Unsupervised learning** refers to analysis goals without training data for the evaluation of the analysis goals. In other words, we have a data set with instances and variables, but there are no specific labels we want to “predict” or learn.

We have only observed input variables, no output variables.

We want to search for interesting patterns

- Frequent itemset and association rule mining
- Sequence mining
- Clustering

Unsupervised techniques often offer a basis or starting point towards supervised ones:

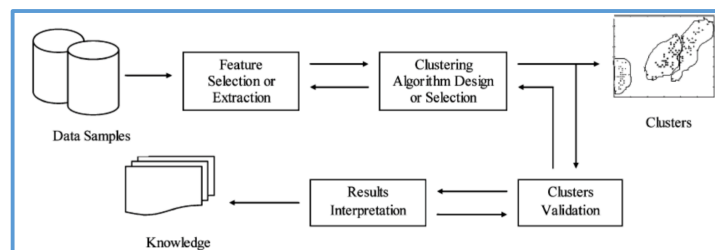
- Predict the discovered patterns, or
- Use the discovered patterns as features in other predictive models

### 2. Clustering

Goal: Finding a grouping of the observations, so-called clusters, that can be used later on for explaining the structure of the observations in the context of the domain.

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups (clusters).

It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.



- Group together similar objects (customers, products, ...)
- Find patterns in data
- Reduce complexity

#### 2.1 Hierarchical Clustering

Goal: create a hierarchical decomposition of the set of objects using some criterion.

We define a hierarchy tree for the observations. Each node in the tree represents a possible subset (cluster) of the observations, the root defines one cluster containing all observations, and the leafs of the tree are the observations representing  $N$  different clusters. Cutting the tree at a certain level provides a possible cluster solution.

In order to decide which clusters should be combined (for *agglomerative*) or where a cluster should be split (for *divisive*), a **measure of (dis)similarity** between sets of observations is required.

In most methods of hierarchical clustering, this is achieved by use of an appropriate metric a measure of distance between pairs of observations), and a linkage criterion which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

A distance or dissimilarity function on a data set  $X$  is defined to satisfy the following conditions:

- 1) Symmetry.  $D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_j, \mathbf{x}_i)$ ;
  - 2) Positivity.  $D(\mathbf{x}_i, \mathbf{x}_j) \geq 0$  for all  $\mathbf{x}_i$  and  $\mathbf{x}_j$ .
- If conditions
- 3) Triangle inequality.
- $$D(\mathbf{x}_i, \mathbf{x}_j) \leq D(\mathbf{x}_i, \mathbf{x}_k) + D(\mathbf{x}_k, \mathbf{x}_j) \text{ for all } \mathbf{x}_i, \mathbf{x}_j, \text{ and } \mathbf{x}_k$$
- and (4) Reflexivity.  $D(\mathbf{x}_i, \mathbf{x}_j) = 0$  iff  $\mathbf{x}_i = \mathbf{x}_j$  also hold, it is called a metric.

Likewise, a similarity function is defined to satisfy the conditions in the following:

- 1) Symmetry.  $S(\mathbf{x}_i, \mathbf{x}_j) = S(\mathbf{x}_j, \mathbf{x}_i)$ ;
  - 2) Positivity.  $0 \leq S(\mathbf{x}_i, \mathbf{x}_j) \leq 1$ , for all  $\mathbf{x}_i$  and  $\mathbf{x}_j$ .
- If it also satisfies conditions
- 3)
- $$S(\mathbf{x}_i, \mathbf{x}_j)S(\mathbf{x}_j, \mathbf{x}_k) \leq [S(\mathbf{x}_i, \mathbf{x}_j) + S(\mathbf{x}_j, \mathbf{x}_k)]S(\mathbf{x}_i, \mathbf{x}_k)$$
- for all  $\mathbf{x}_i, \mathbf{x}_j$  and  $\mathbf{x}_k$
- and (4)  $S(\mathbf{x}_i, \mathbf{x}_j) = 1$  iff  $\mathbf{x}_i = \mathbf{x}_j$ , it is called a similarity metric.



Measures	Forms	Comments	Examples and Applications
Minkowski distance	$D_p = \left( \sum_{j=1}^n  x_{ij} - x_{ij} ^p \right)^{1/p}$	Metric. Invariant to any translation and rotation only for $n=2$ (Euclidean distance). Features with large values and variances tend to dominate over other features.	Fuzzy c-means with measures based on Minkowski family [130]
Euclidean distance	$D_2 = \left( \sum_{j=1}^n  x_{ij} - x_{ij} ^2 \right)^{1/2}$	The most commonly used metric. Special case of Minkowski metric at $n=2$ . Tend to form hyperspherical clusters.	K-means algorithm [191]
City-block distance	$D_1 = \sum_{j=1}^n  x_{ij} - x_{ij} $	Special case of Minkowski metric at $n=1$ . Tend to form hyperrectangular clusters.	Fuzzy ART [57]
Sup distance	$D_\infty = \max_{j=1, \dots, n}  x_{ij} - x_{ij} $	Special case of Minkowski metric at $n \rightarrow \infty$ .	Fuzzy c-means with sup norm [29]
Mahalanobis distance	$D_\theta = (x_i - x_j)^T S^{-1} (x_i - x_j)$ , where $S$ is the within-group covariance matrix.	Invariant to any nonsingular linear transformation. $S$ is calculated based on all objects. Tend to form hyperellipsoidal clusters. When features are not correlated, squared Mahalanobis distance is equivalent to squared Euclidean distance. May cause some computational burden.	Ellipsoidal ART [13], Hyperellipsoidal clustering algorithm [194]
Pearson correlation	$D_p = (1 - r_p)/2$ , where $r_p = \frac{\sum_{j=1}^n (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_j)}{\sqrt{\sum_{j=1}^n (x_{ij} - \bar{x}_i)^2 \sum_{j=1}^n (x_{ij} - \bar{x}_j)^2}}$	Not a metric. Derived from correlation coefficient. Unable to detect the magnitude of differences of two variables.	Widely used as the measure for analyzing gene expression data [80]
Point symmetry distance	$D_p = \min_{i=1, \dots, n} \left[ \frac{\ x_i - x_j\  + \ x_j - x_i\ }{2} \right]$	Not a metric. Compute the distance between an object $x_i$ and a reference point $x_j$ . $D_p$ is minimized when a symmetric pattern exists.	SIBKM (Symmetry-based K-means) [264]
Cosine similarity	$S_\theta = \cos \alpha = \frac{x_i^T x_j}{\ x_i\  \ x_j\ }$	Independent of vector length. Invariant to rotation, but not to linear transformations.	The most commonly used measure in document clustering [261]

In case of **quantitative variables**, the most important distance is the Euclidean distance.

$$d^2(x, z) = \sum_{j=1}^p (x_j - z_j)^2 = \|x - z\|^2.$$

In the case of **qualitative variables**, the most frequently used distance is the Hamming distance. This distance uses dummy coding for the different values of the qualitative variable, i.e., each value corresponds to a dummy variable with the values 0 and 1.

Other distance metrics exist

- **Levensthein** distance: distance between sequences
- **Jaccard**: measures dissimilarity between sample sets

$$d_J(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

Measuring the distance between observations with variables of different magnitude puts an emphasis on the distance between the variables with larger scale. Hence, it is usually recommended to standardize the variables into a value range 0; 1. Such standardization is of utmost importance if we want to measure the distances between observations with qualitative and quantitative variables.

### 2.1.1 Agglomerative

⇒ Starting with single elements and aggregating them into clusters.

- **Single linkage**: minimum of object distances
  - Start with one data point (object) per cluster
  - Join two clusters if the minimal distance between two objects of both clusters is minimal: long clusters
- **Complete linkage**: maximum of object distances
  - Start with one data point (object) per cluster
  - Join two clusters if the maximal distance between two objects of both clusters is minimal: tight clusters

- **Average linkage:** join two clusters if the mean distance between all observations in both clusters is minimal
- **Ward method:** calculate per pair of potential clusters the total within-cluster sum of squares if they would be merged. Join two clusters if their combined within-cluster SS is minimal.

---

**Algorithm 4:** Agglomerative clustering
 

---

```

1 Define clusters  $C_k, 1 \leq k \leq N$  by the observations,  $N_{cl} = N$ ;
2 for  $k = 1$  to  $N - 1$  do
3   Merge clusters  $C_r$  and  $C_s$  for which  $d(C_r, C_s) = \min_{(l,k)} D((C_l, C_k))$ ;
4    $N_{cl} = N_{cl} - 1$ ;
5 end
  
```

---

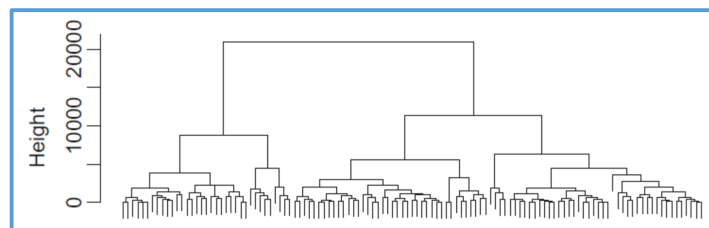
### 2.1.2 Divisive

⇒ Starting with the complete data set and dividing it into partitions

Start with all data points in one cluster. Separate clusters iteratively using some cluster algorithm until some stopping criterion is fulfilled.

This is more efficient than hierarchical agglomerative clustering if not clustered “all way down”.

**Dendrograms** are a common visualization method for hierarchical clustering. The length of branches corresponds with distance between merged objects. It can be used to determine the number of clusters (but better methods exist, cfr. Infra.).



Merging two clusters is not advisable if the height of the branches measured from the merged cluster to the individual clusters is a substantial value. The evaluation of distances is easier using a **scree plot**, which shows the distance between the clusters plotted against the number of clusters.

## 2.2 Partitional Clustering

Goal: Construct various partitions and then evaluate them by some criterion.  
k-means; fuzzy c-means; rough k-means; possibilistic c-means; ...

As opposed to hierarchical clustering, this method defines a number of clusters in advance and assigns the observations iteratively to the clusters.

Partitional clustering is non-hierarchical, each instance is placed in **exactly** one of  $K$  non-overlapping clusters. Since only one set of clusters is output, the user normally has to input the desired number of cluster  $K$ .

### 2.2.1 K-Means

⇒ Most well-known Partitional clustering

- Good for large datasets; simple
- Can be parallelized
- Not robust with respect to outliers. Indeed, outliers can be interpreted as small groups of observation with rather irregular behaviour.
- Initialization is important!
- Not suitable for categorical data (mean needs to be calculated)

---

#### Algorithm 5: $K$ -means algorithm

---

**Data:** Observation matrix  $\mathbf{X}$  and distance for the objects; number of clusters  $K$ .

**Result:** Cluster solution for observations

```

1 begin
2   Define an initial solution for the cluster centers  $(c_1, c_2, \dots, c_k)$ ;
3   Assign each observation  $x$  to the cluster which center is closest to the
   observation;
4   Compute new centers for the clusters as means of the assigned
   observations;
5   Repeat steps 2 and 3 as long as there is no significant change in the centers;
6 end
```

---

### 2.2.2 PAM (Partitioning Around Medoids)

This method is based on Medoids instead of means.

Medoid: multidimensional generalization of the concept of the median, i.e., a central point in the multivariate data. It is an existing data point.

- Less sensitive to outliers than  $k$ -means
- Also works for categorical data, no calculation of mean
- Iterative algorithm: start from set of Medoids (existing observations in the dataset)
  - 1) **Assignment step:** allocate observations to closest Medoid
  - 2) **Swapping step:** find cluster observations with smaller distances to other cluster observations than current Medoid
  - 3) **Calculation step:** calculate new Medoid.

### 2.3 Model-based clustering

This method estimates a **mixture model** for the distribution of observations. It assumes a fixed number of subpopulations/clusters from which observations are drawn, each with its own distribution.

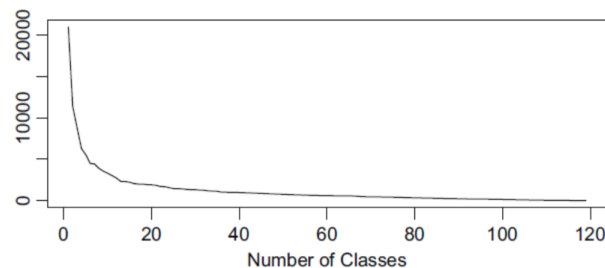
The algorithm tries to estimate the different parameters of the subpopulations' distributions according to a model selection criterion (e.g. Expected-Maximization). Then, assign observations to most likely distribution by selecting cluster whose density is the highest.

Ex:

- Assume clusters follow a normal distribution
  - Estimate mean and covariance of each distribution + the proportions using the EM algorithm
  - Calculate per observation the densities for the different distributions
  - Assign observation to cluster with highest density
- Relatively efficient and robust
  - Also works for categorical data
  - However, initialization is still important!

### 3. Number of clusters

- Visualization: scree plot
  - Distance between clusters vs. Number of clusters
  - Look at “elbow” in the plot



- Stopping rules
- Optimizing certain criteria
  - Akaike's Information Criterion (AIC)
  - Bayesian Inference Criterion (BIC)
- Heuristics

### 4. Applications

- Market research: market segments, product positioning
- Social network analysis: recognize communities of people
- Social science: identify students, employees with similar properties
- Search: group search results
- Recommender systems: predict preferences based on user's cluster

- Crime analysis: identify areas with similar crimes
- Image representation and colour palette detection

## 5. Validation

Many indices; e.g. (Halkidi, Batistakis, Vazirgiannis 2001). These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters

Davis Bouldin Index (Davies and Bouldin, 1979):

$$DB_{nc} = \frac{1}{nc} \sum_{i=1}^{nc} R_i$$

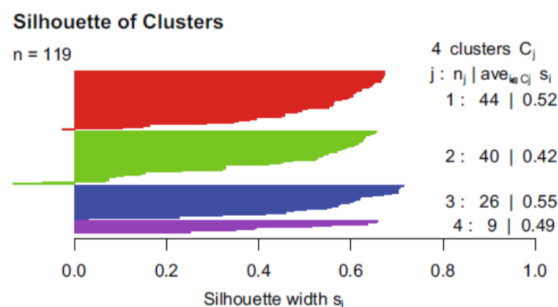
$$R_i = \max_{j=1, \dots, nc, i \neq j} R_{ij}, \quad i = 1, \dots, nc$$

$$R_{ij} = (s_i + s_j) / d_{ij}.$$

- $s_i$  = mean distance of all objects of one class to its centre.
- $d_{ij}$  = distance between centres of classes  $i$  and  $j$
- A good segmentation is characterized by  $DB_{nc}$  low

## Silhouette

- Plot for each observation its silhouette value
- Silhouette values are based on
  - Average distance from each observation to observations within cluster
  - Minimum distance from each observation to observations outside of cluster
- Large positive values indicate good clustering assignment for that observation
- Negative values indicate bad clustering
- Silhouette coefficient: overall quality measure of clustering solution



## CHAPTER 9: KNOWLEDGE REPRESENTATION & REASONING

In KBS, the knowledge is made explicit, rather than being implicitly mixed in with the algorithm. When this is done, the algorithm also has to include a reasoning or inference mechanism so that deductions and conclusions can be drawn from the knowledge.

With KBS approaches, the successful automation of knowledge-intensive tasks is dramatically increased.

### 1. Frame Based Systems

Recall, **frame systems** attempt to reason about classes of objects by using **prototypical representations** of knowledge which hold good for the majority of cases but which may need not be deformed in some way to capture the complexities of the real world.

Frame based systems are more powerful, but also more complex and more difficult to develop than simpler object-attribute-value/rule systems.

#### 1.1 Frames

Frame: stereotypical knowledge – knowledge of some concept

A **frame** is a description of an object that contains slots for all of the information associated with the objects. Slots, like attributes, may store values. Slots may also contain default values, points to other frames, sets of rules, or procedures by which values may be obtained. The inclusion of these additional features make frames different from object-attribute-value triplets.

CLASS FRAME			INSTANCE FRAME		
Frame Name	Bird		Frame Name	Tweety	
Properties	Color	unknown	Properties	Color	yellow
	Eats	Worms		Eats	worms
	#wings	2		#wings	1
	Flies	true		Flies	False
	Hungry	unknown		Hungry	unknown
	Activity	unknown		Activity	unknown

Frames hold the structure and behaviour of a class of objects/concepts.

- Structure: properties/data/attributes
- Behaviour: methods, rules defining how instances of the class typically behave

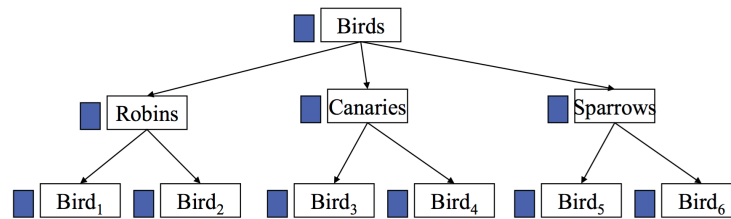
→ Generic structure and behaviour defined at class level is specified in instances.

#### 1.2 Inheritance

Structural and behavioural properties are inherited from super-class to sub-class. That is to say, subclasses automatically carry the structure and behaviour of super-class.

>< Exception handling. Ex: Birds fly, but penguins do not.

Multiple inheritance



### 1.3 Facets

Facets provide further information on a property/attribute

- Define constraints on property value: data type, range of values, etc.
- How to obtain a value for a property: if-needed method
- What to do if a value changes: if-changed method

Slot	Filler
NAME	Plot_of_land WHEN ADDED perform Procedure <i>Ask_place_&amp;_Size</i>
PLACE	Perform Procedure <i>Validate</i>
SIZE	WHEN UPDATED perform Procedure <i>Determine_Value</i>
PRICE	WHEN NEEDED perform Procedure <i>Determine_Value</i>

Object name: Refrigerator  
Class: Home Appliances

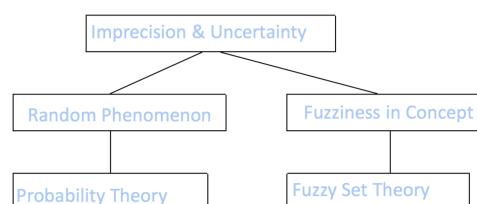
Age	10
Service#	123-1998
Volume	16
Condition	Good
Location	Kitchen
Power	Electricity
Defrost cycle	ON

- Initialize
- from Database
- from Procedure
- Expert Rules
- User specified
- by Inheritance
- Message passing from other objects

## 2. Fuzzy Sets and Reasoning

**Fuzzy set:** membership of a set. You just want to give a membership value to an element in a set. Ex: How well do you speak English? It is not a 0 or 1. It is a matter of membership.

It is basically knowledge theory. If you know a fact which is fuzzy and rule (not fuzzy), you can still make some conclusions.



Examples of fuzzy (linguistic) terms:  
most, more-or-less, much more than, tall, good, very good, very tall,  
approximately, about, nearly, .....

In daily life, people have to think and reason with imprecisions & uncertainty in many cases.

If x is A then y is B (where A & B are linguistic values defined by fuzzy sets on universes of discourse X & Y).

- "x is A" is called the **antecedent** or premise
- "y is B" is called the **consequence** or conclusion

Examples:

- If pressure is high, then volume is small.

- If the road is slippery, then driving is dangerous.
- If a tomato is red, then it is ripe.
- If the speed is high, then apply the brake a little.
- A fuzzy if-then rule can be viewed as a fuzzy relation R.

**Fuzzy reasoning**, also known as **approximate reasoning**, is an inference procedure that derives conclusions from a set of fuzzy if-then-rules & known facts.

Given A,  $A \Rightarrow B$ , infer B (here still without fuzzy)

A = "today is sunny"

$A \Rightarrow B$ : day = sunny then sky = blue

infer: "sky is blue"

#### Approximation

A' = "today is more or less sunny"

B' = "sky is more or less blue"

illustration

Premise 1 (fact): x is A'

Premise 2 (rule): if x is A then y is B

---

Consequence: y is B'

(approximate reasoning or fuzzy reasoning!)

#### Definition of fuzzy reasoning

Let A, A' and B be fuzzy sets of X, X, and Y, respectively.

Assume that the fuzzy implication  $A \Rightarrow B$  is expressed as a fuzzy relation R on  $X \times Y$ .

Then the fuzzy set B induced by "x is A'" and the fuzzy rule "if x is A then y is B" is defined by:

$$\mu_{B'}(y) = \max_x \min[\mu_{A'}(x), \mu_R(x, y)]$$

#### Single rule with single antecedent

Rule : if x is A then y is B

Fact: x is A'

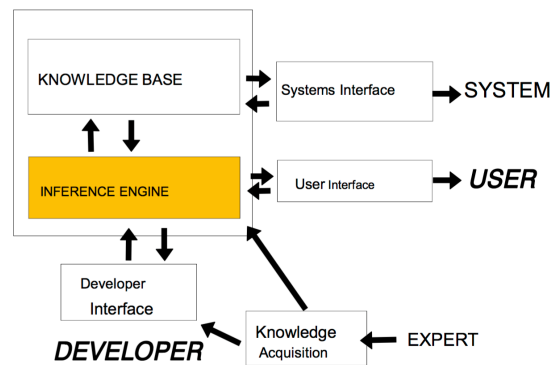
---

Conclusion: y is B'

### 3. Inference and Reasoning



## 3.1 Recall



**The inference engine** deals with the rules in a number of ways. We have already seen forward and backward chaining.

- 1) Knowledge is separated from the inference and control
- 2) Different ways to represent knowledge
- 3) Heuristic search

**Forward chaining** starts from the facts/available information. It then applies rules that match. Repeat until nothing more can be concluded.

When?

- All or most of the data are given in the initial problem statement
- There are a large number of potential goals, with only a few ways to use the given facts
- It is difficult to formulate a goal or hypothesis

R1: IF A and C THEN E  
 R2: IF D and C THEN F  
 R3: IF B and E THEN F  
 R4: IF B THEN C  
 R5: IF F THEN G

Given facts:

A is true  
 B is true

What can be concluded?

**Backward chaining** starts from the goal. It works towards the facts. Apply only rules that are relevant to the (sub)goal. Attempts to prove a goal by gathering information

When?

- A goal or problem statement is given or can easily be formulated
- There are large number of rules that match the facts of the problem and thus produce a large number of conclusions or goals
- Problem data are not given but must be acquired. Goal driven search can help guide data acquisition.

R1: IF B and C THEN G  
 R2: IF A and G THEN I  
 R3: IF D and G THEN J

R4: IF E or F THEN C  
 R5: IF D and C THEN K  
 Goal I

⇒ Which strategy is best

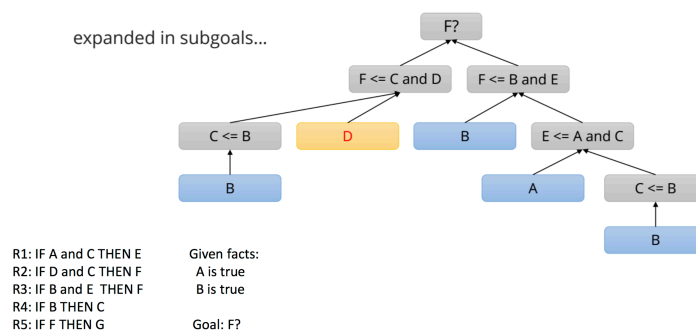
Consider the problem of confirming or denying the statement "I am a descendant of Napoleon Bonaparte". A solution path is a direct line between the "I" and Napoleon. This space may be

searched in each direction, starting with the "I" and working along ancestor lines to Bonaparte or starting with Bonaparte and working through his descendants.

Napoleon was born about 200 years ago. If one assumes 20 years per generation; the required path will be of length 10. Since each person has exactly two parents, search back from the "I" would examine about 210 ancestors.

A search that worked forward from Napoleon would examine more states since people tend to have more than two children. If one assumes an average of three children per family, the search would examine about 310 nodes of the family tree.

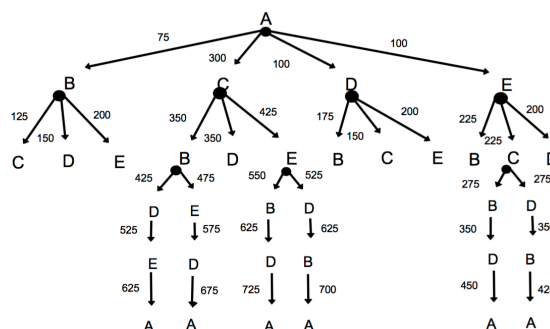
### 3.2 Searching the goal state space



Many search problems can be explored using states and transitions. Ex: Traveling Salesman Problem.

State: where we are in the search. Derive possible transitions from here and proceed.

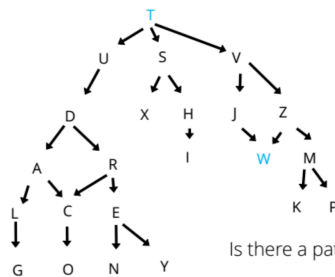
Transitions: ways to go to the next state.



We go through the state space, looking for the shortest path.

- **Breadth-first**: stay as long as possible on the same level. It is more interesting as it gives general questions first.  
This method always keeps track of other alternatives.

Breadth-first:



Is there a path T → W?

TODO

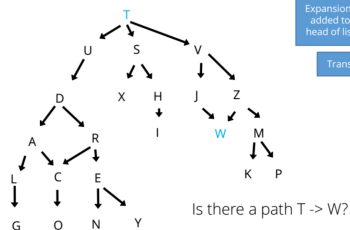
- {T}
- {U,S,V}
- {S,V,D}
- {V,D,X,H}
- {D,X,H,J,Z}
- {X,H,J,Z,A,R}
- {J,Z,A,R,I}
- {Z,A,R,I,W}

SEEN

- {T}
- {T,U}
- {T,U,S}
- {T,U,S,V}
- {T,U,S,V,D}
- {T,U,S,V,D,X,H}
- {T,U,S,V,D,X,H,J}
- {T,U,S,V,D,X,H,J,W}

- **Depth-first:** go from top to bottom as soon as possible. We will put the children of an element in front of the list. We try to go down as soon as possible by expanding all the children we then end up in a leaf node.

Depth-first:



Is there a path T → W?

TODO

- {T}
- {U,S,V}
- {D,S,V}
- {A,R,S,V}
- {L,C,R,S,V}
- {G,C,R,S,V}
- {O,R,S,V}
- {E,S,V}
- {N,Y,S,V}
- {X,H,V}
- {I,V}
- {J,Z}
- {W,Z}

SEEN

- {T}
- {T,U}
- {T,U,D}
- {T,U,D,A}
- {T,U,D,A,L}
- {T,U,D,A,L,G,C}
- {T,U,D,A,L,G,C,O,R}
- {T,U,D,A,L,G,C,O,R,E}
- {T,U,D,A,L,G,C,O,R,E,N,Y,S}
- {T,U,D,A,L,G,C,O,R,E,N,Y,S,X,H}
- {T,U,D,A,L,G,C,O,R,E,N,Y,S,X,H,I,V}
- {T,U,D,A,L,G,C,O,R,E,N,Y,S,X,H,I,J}
- {T,U,D,A,L,G,C,O,R,E,N,Y,S,X,H,I,J,W}

STATE

### 3.2.1 Hill Climbing Algorithm

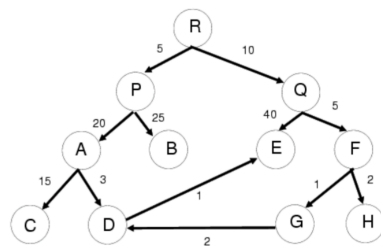
⇒ Repeat until all the nodes in the search space have been examined

- 1) Check the head of the list of nodes under consideration. If it is the goal state report the success and stop, otherwise continue.
- 2) Sort the children of the head of the list (if it has any)
- 3) Replace the head with the sorted list of its children (if it has any)

### 3.2.2 The Best First Search Algorithm

⇒ Repeat until all the nodes in the search space have been examined

- 1) Check the head of the list of nodes under consideration. If it is the goal state report the success and stop, otherwise continue.
- 2) Replace the node at the head of the list with its children
- 3) Sort the entire list



Find path from R→E

## ▪ Hill-climbing:

- {R}
- {R-P (5), R-Q (10)}
- {R-P (20), P-B (25), R-Q}
- {A-D (3), A-C (15), P-B, R-Q}
- {D-E (1), A-C, P-B, R-Q}
- RESULT: (R, P, A, D, E), cost:  $5 + 20 + 3 + 1 = 29$

Expansions added to head but sorted on cost (Greedy depth first)

## ▪ Best-first:

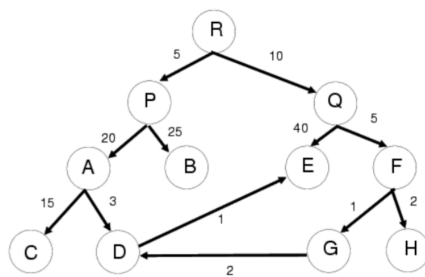
- {R}
- {R-P (5), R-Q (10)}
- {R-Q (10), P-A (20), P-B (25)}
- {Q-F (5), P-A (20), P-B (25), Q-E (40)}
- {Q-F (1), F-H (2), P-A (20), P-B (25), Q-E (40)}
- {G-D (2), F-H (2), P-A (20), P-B (25), Q-E (40)}
- {D-E (1), F-H (2), P-A (20), P-B (25), Q-E (40)}
- RESULT: (R, Q, F, G, D, E), cost:  $10 + 5 + 1 + 2 + 1 = 19$

Whole list sorted on cost after every expansion

## 3.2.3 Branch and Bound Search

⇒ Repeat until all the nodes in the search space have been examined

- 1) Check the head of the list of paths to see if it leads to the goal state. If it does, report the success and stop, otherwise continue.
- 2) Extend the node at the head of the list of paths by one level. This will result in creating many new paths.
- 3) Sort all the paths in the list by their accumulated cost so that the least-cost path will be at the head of the list



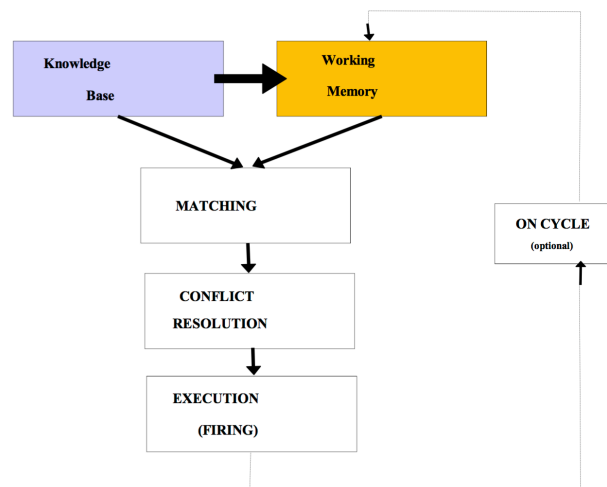
Find path from R→E

## ▪ Branch and bound:

- {R}
- {R-P (5), R-Q (10)}
- {R-Q (10), R-P-A (25), R-P-B (30)}
- {R-Q-F (15), R-P-A (25), R-P-B (30), R-Q-E (50)}
- {R-Q-F-G (16), R-Q-F-H (17), R-P-A (25), R-P-B (30), R-Q-E (50)}
- {R-Q-F-H (17), R-Q-F-G-D (18), R-P-A (25), R-P-B (30), R-Q-E (50)}
- {R-Q-F-G-D (18), R-P-A (25), R-P-B (30), R-Q-E (50)}
- {R-Q-F-G-D-E (19), R-P-A (25), R-P-B (30), R-Q-E (50)}
- RESULT: (R, Q, F, G, D, E), cost:  $10 + 5 + 1 + 2 + 1 = 19$

Expansions and sort list of complete paths

## 3.3 The inference cycle



- 1) **Knowledge base** contains the knowledge

- 2) **Working memory** is empty in between two executions of an application. When the application is started, the known facts are stored in working memory. They will be updated by the inference engine with a rule interpreter which determines which rules are applicable for the current problem.
- 3) **The matching algorithm** compares the contents of working memory to facts and rules contained in the knowledge base
  - **Forward chaining:** compares facts in working memory with the premises of rules in the knowledge base
  - **Backward chaining:** compares goals to conclusions of rules

Exhaustive matching of all rules and their conditions or conclusions against working memory may require a huge number of comparisons. An efficient matching algorithm: RETE, avoids repetitive matching by indexing.

**RETE algorithm:** is a pattern matching algorithm for implementing production rule systems. It is used to determine which of the system's rules should fire based on its data store.
- 4) **Conflict resolution:** multiple rules may be eligible (triggered) for execution. Among these rules, one has to be chosen. The mechanism to choose between candidate rule (the conflict set) is called conflict resolution.
- 5) **Execution:** The selected rule is executed; Everything runs in cycle until we have reached the goal. After every cycle we can perform some checks. Updates data in working memory; matching changes are updated.
  - Refractoriness (fire only once on same data); Priorities; Most Recently Used; Lest Recently Used; Most Specific First (more premises); Lexicographical Order
- 6) **On-Cycle:** after every cycle this can be executed: perform checks; user interaction; Demon rules

#### Examples of Common KBS applications

- Machine fault diagnosis
- Process control
- Equipment Configuring
- Maintenance planning
- Quality control
- Risk assessment
- Medical Diagnosis
- Intelligent Monitoring
- Demand Forecasting
- Task Scheduling
- Process Design
- Advising on safety

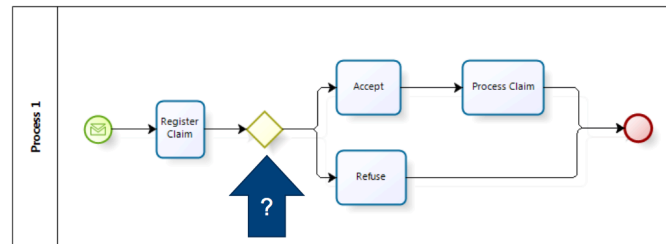
- financial analysis
- software support systems

## CHAPTER 10: DECISION ANALYTICS

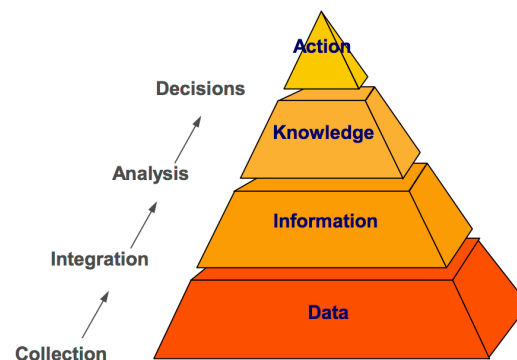
⇒ How does decision mining relate to normal analytics?

### 1. Framing Analytics decisions

Decisions are important for business. Why would we only care about/model the data or the processes? Where is the decision? How is the decision logic modelled? BPM tries to find decisions and model them.



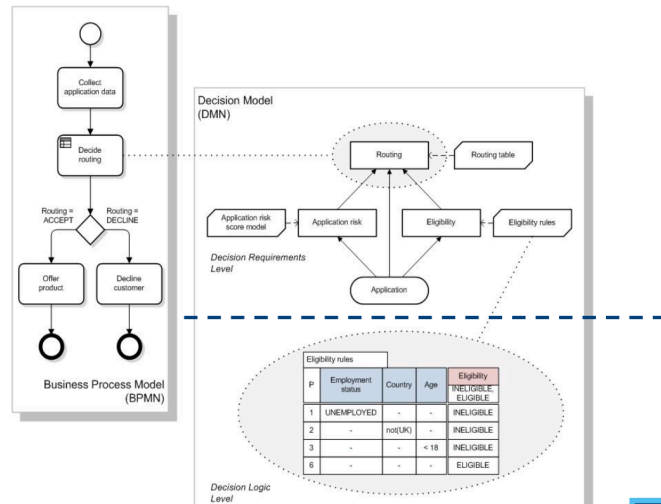
The normal decision making is often based on large amounts of data. It is important to be able to manage the decision: from **big data to action**.



- What is the decision? Eligibility, price, insurance, theft rating, customer offer, retention, supplier selection, hire, credit, etc.
- What, what how?
- Who owns the decision?
- Who makes the decision every day?
- Who is impacted?
- What triggers the decision?
- When are we making the decision?
- What is required to make this decision?
  - Information requirements
  - Knowledge sources (regulations, analytics, expertise)
  - Other decisions
  - Decision logic

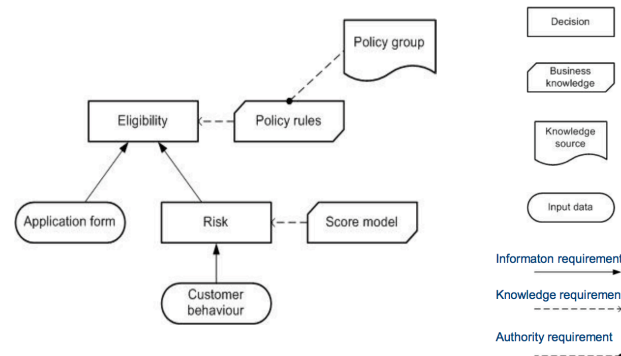
Decision(s) (logic) need to be modelled. A standard for processes (BPMN) is not enough. A standard for case management (CMMN) is not enough.

## → Decision Modelling & Notation Standard (DMN)



In DMN there are two levels

- **Decision requirements level:** single decision requirement graph depicted as a set of decision requirement diagrams. Decision requirement diagrams (DR) denote the information requirements of each decision, by connecting them with their sub decisions and inputs. This is represented by a directed acyclic graph. The DMN specification allows a DRD to be an incomplete or partial representation of the decision requirements in a decision model.



- **Decision logic level:** A decision is the description of the decision logic used to determine an output from a number of inputs.

### Decision requirement graph

The decision requirement graph expresses the top level, then requirements. It does not tell us what to do, it does not set up a procedure but only details the requirement.

Ex: Selecting a marketing offer depends on 3 other decisions. By drawing a model of these decisions we get more insights. What are sources, big areas, etc. Try to figure out about related decisions.

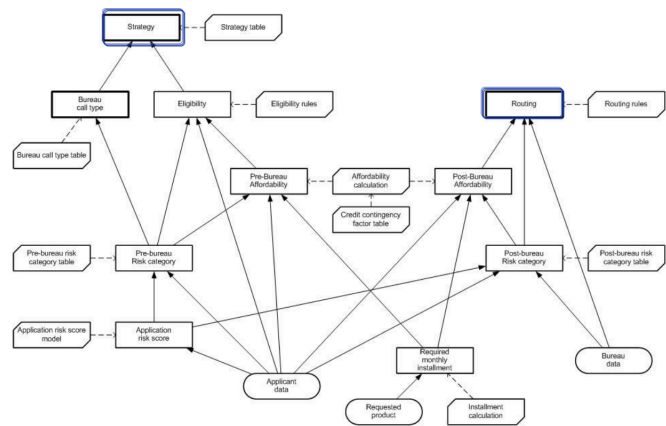
- **Property 1:** Given a decision model every decision D in that model has a unique decision requirement graph  $DRG_D$  with D as its single top-level decision.



- **Property 2:** The topological order of a DRD induces a partial order  $\leq$  on the decisions contained in the DRD.

For two decisions D1 and D2 we say  $D2 \leq D1$  if and only if there is a directed path from D2 to D1, i.e. D2 is a sub decision of D1.

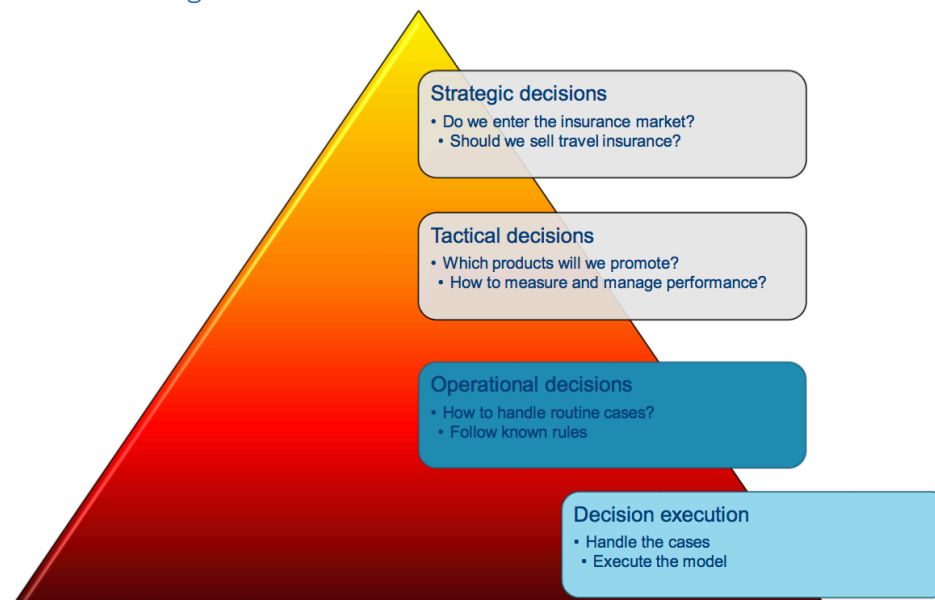
Since decisions are declarative, this partial order does not indicate an execution order, but rather a requirement order.



## Modelling Decision Logic

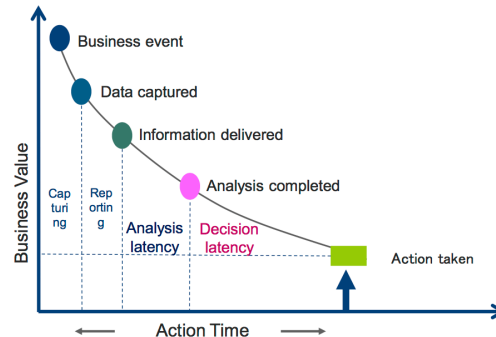
Good decision table models are a proven technique to represent decision rules. Consistency, completeness and correctness by design.

## 2. Decision management



**Operational decisions** are simple day-to-day decisions. These decisions are made **frequently, rapidly, consistently** and in **high volume**. Ex: go for loan or credit. They are

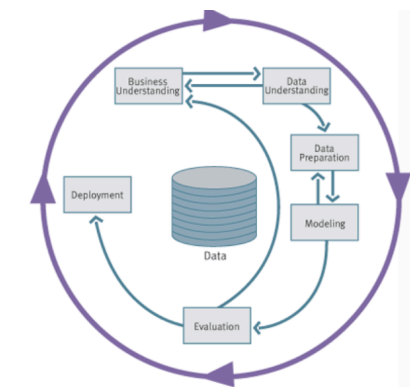
frequent and non-trivial, next, they are rather deterministic. However, operational decisions might change frequently and require some explanation. Routine decisions then do be **automated**, it could however take some time to automate, collect data, and decide.



### Decisions Management & Analytics

#### SEMMA

- S: Sample (Training, Validation, Test)
- E: Explore (get an idea of the data at hand)
- M: Modify (select, transform)
- M: Model (create data mining model)
- A: Assess (validate model)



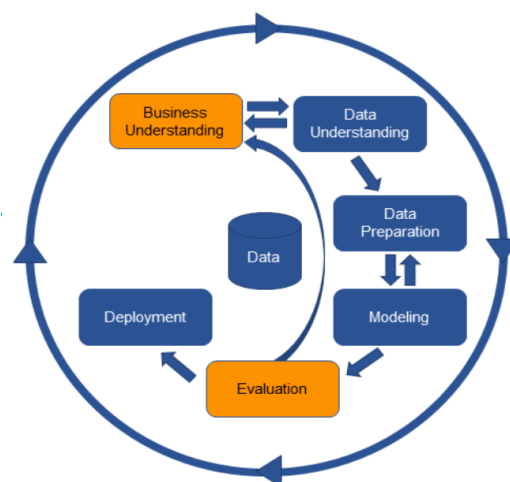
Business Understanding and evaluation are the link with decision making.

- **Understanding:** analytics will not, by itself, improve the business (churn, metrics, retention, ...). Decision-making does.

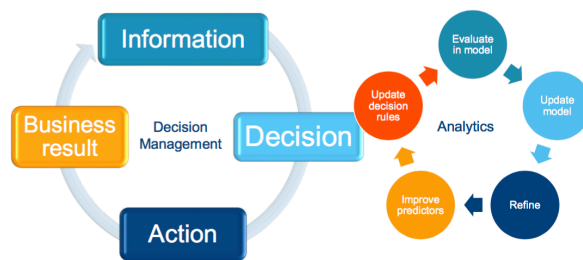
Which decisions must change, how are these decisions made, and by whom?

*"Successful analytic teams spend more time understanding the business problem and less time wading through lakes of data."*

- **Evaluation:** the analytic measures (accuracy, lift, ...) must be put into the broader business context to see how that will impact overall decision-making. *"Successful analytic teams minimize the white space between analytic success and business success, evaluating their models against business drivers as well as analytic ones."*



## Integrating analytics



We have a cycle about building model, update the model and improve it.

Managing decisions:

“... Provide a common notation that is readily understandable by all business users ... DMN creates a standardized bridge for the gap between the business decision design and decision implementation.”

- Information required
- Decision knowledge
- Context
- Business goals
- Motivation
- Human Decision Making
- Discovery
- Implementation

Decisions and process: the role of decision models

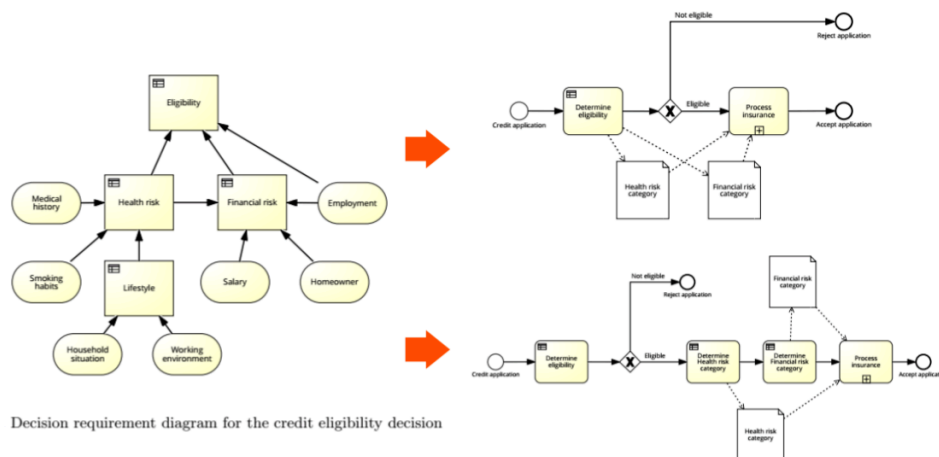
### 1) A Decision Model corresponding to a single Decision Activity in a Process Model

When there is no interleaving of data throughout the model, decisions can be used in this form of single branches that use the output of an activity that has implemented and evaluated the criteria for making the decision.

### 2) A decision Model Spanning over Multiple Decision Activities in an existing Process Model

Multiple activities in a process model may refer to different decisions that are all part of the same decision model. Are the data needed as input for a decision available before the decision is invoked? Activities producing the input required by decisions should occur before invoking those decisions. Otherwise, situations appear in which decisions cannot be made consistently due to incomplete or incorrect input.

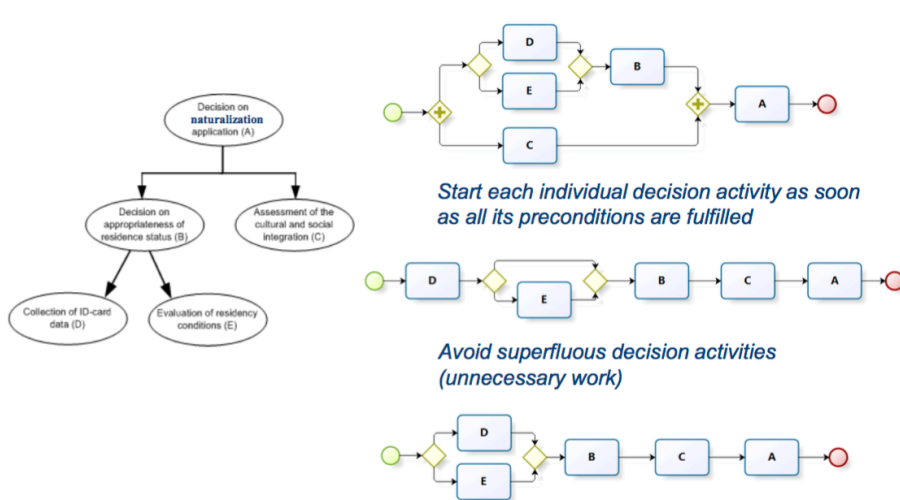
When decisions are dependent on intermediary results that are spawned earlier in the process, the decision model also puts constraints on the sequence ordering of the workflow.



### 3) A Decision Model that can be translated to a straightforward Process for Execution

Sometimes the business process is really about a big decision. Some approaches model this in business process models, hence forgoing the purpose of decision models that were designed specifically for this task.

Other approaches rather seek to find the balance between data-driven models and business processes. However, the process part still remains a subordinate to the decision model. When the process is really about a big decision, the process model can be considered as the chosen execution flow to make the decision.




From decisions to processes

### 4) Executing a Decision Model Beyond One Fixed Decision: Flexibility

Once a decision model is built, it could be used for multiple purposes, not just the obvious decision that is present in the context of a current business question. The decision model could be designed for the current process, but also for other or future processes.

- The basic decision/process:
  - John Doe is 50 and has 30 years of service, process John's case
  - The customer puts in a claim/order/loan request.
  - Do we accept the order? What is the price? How to process the order, etc.
- There is more than this basic decision
  - What-If, Optimization and Incomplete data

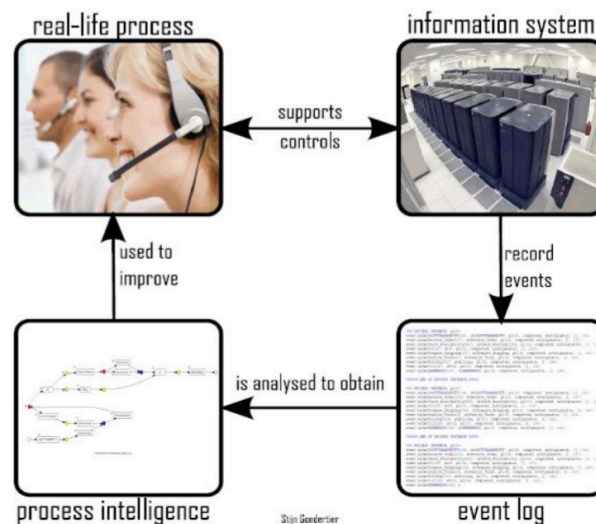
- Why do I get this result?
- What can I already decide with these incomplete data?
- Goal seeking: What do I have to change to obtain that result?  Optimization: How do I get the maximum?
- Overview: Who finally gets this result? Where is my case?
- Decision Analysis: Are there some strange assignments in giving this result?
- Decision Maintenance: What if the policy changes?
- Traceability: How easy is it to trace back to the original text (rules)?
- What is the decision?

If we look at decisions and data mining, we observe these to come quite together. Once we are talking about decisions and processes, it becomes easy to build a model.

Very often there are some complicated issues. If you need a result higher in the process, you cannot simply hide this. If you have multiple activities, they may refer to many decisions. If you split decisions in sub-decisions and spread decisions to your process, you have to fix the order of decisions.

Split: each of sub-decisions receive data needed to make decisions. If we want to spread down a little, we have to be **consistent** and make sure every decision has access to data needed. Consistent in sense that I can only make decision if data is available and if I am sure data and decisions can come back.

### 3. Decision mining



**Data mining** (Ex: predictive analytics). I have a big data set with labels and attributes.

Typical data mining log:

- Single occurrence per case;
- Focus on (labelled) data;
- Used for classification, regression, etc.

id	attr1	attr2	attr3	label
1	x1	y6	z3	l1
2	x2	y5	z2	l1
3	x1	y5	z54	l1
4	x1	y4	z7	l2
5	x2	y3	z9	l2

**Process discovery:** summarizes an event log. Process mining tries to use this sequential information.

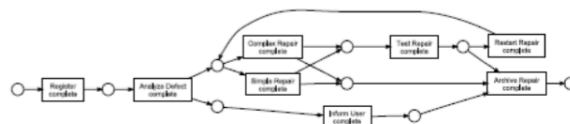
Note: event logs may only contain a subset of all possible traces of a process model. the models produced with process discovery are based on events/behaviour present in the event log.

Process mining basically only contains information about time and resource. It starts from an event.

In reality, we want to build a process model based on event data. In most cases we have:

- Multiple events which are all time stamped
- Extra data;

Case ID	Task Name	Event Type	Originator	Timestamp	Extra Data
1	Register	completed	System	09-01-2008 14:00:00	...
1	Analyze Defect	started	Tester3	09-01-2008 15:00:00	...
1	Analyze Defect	completed	Tester3	09-01-2008 15:01:00	...
1	Complex Repair	started	SolverC1	09-01-2008 15:05:00	...
1	Complex Repair	completed	SolverC1	09-01-2008 15:07:00	...
1	Test Repair	completed	Tester2	09-01-2008 15:07:00	...
1	Archive Repair	completed	System	09-01-2008 15:30:00	...
2	Register	completed	System	09-01-2008 15:35:00	...
2	Analyze Defect	started	Tester3	09-01-2008 18:10:00	...
2	Analyze Defect	completed	Tester3	09-01-2008 18:25:00	...
2	Simple Repair	started	SolverC2	09-01-2008 18:30:00	...
2	Simple Repair	completed	SolverC2	09-01-2008 18:31:00	...
2	Restart Repair	completed	System	09-01-2008 18:59:00	...



Typical process mining log:

- Multiple occurrences/events per case
- Ordered or timestamped
- Data items tied to events are often overlooked
- Focuses on control flow

case id	time	event	resource
1	15:20	A	r1
1	15:21	A	r1
1	16:45	B	r2
2	14:01	A	r1
2	14:58	B	r3
2	15:02	C	r2
3	9:43	A	r2
3	23:19	C	r1

## Decision log:

- Multiple occurrences/Events per case;
- Ordered or timestamped
- Extensive data
  - Labels, timestamps, complex data types, etc.

case id	time	event	attr
1	15:20	A	{{res=r1;attr1=x1};attr2=y6;attr3=z3}
1	15:21	A	{{res=r1;attr1=x2};attr2=y6;attr3=z3}
1	16:45	B	{{res=r2;attr1=x1};attr2=y5;attr3=z54}
2	14:01	A	{{res=r1;attr1=x2};attr2=y4;attr3=z7}
2	14:58	B	{{res=r3;attr1=x2};attr2=y3;attr3=z9}
2	15:02	C	{{res=r2;attr1=x1};attr2=y6;attr3=z54}
3	9:43	A	{{res=r2;attr1=x1};attr2=y5;attr3=z2}
3	23:19	C	{{res=r3;attr1=x2};attr2=y4;attr3=z7}

Traditional data mining is based on data logs. Control flow-based discovery is based on process logs

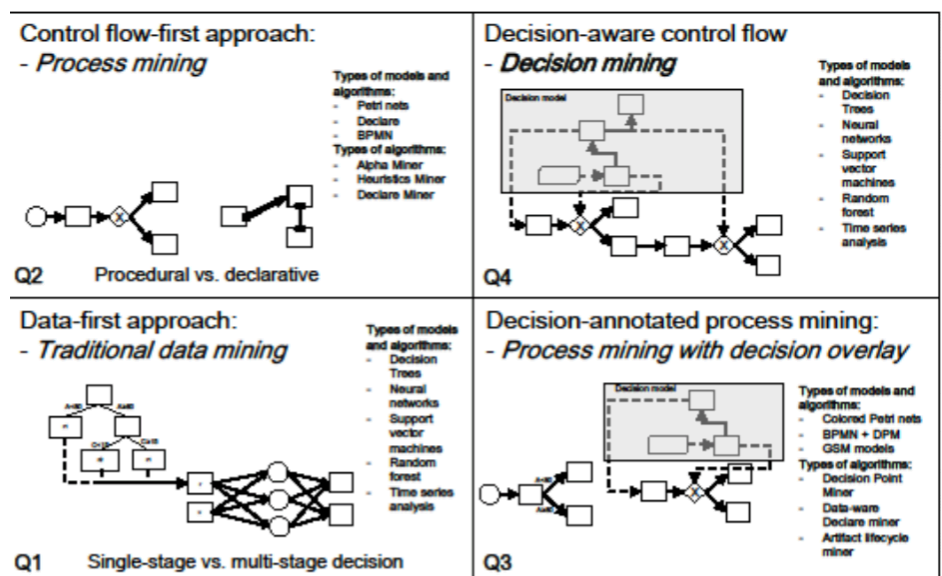
- Heuristics Miner, Inductive Miner, MINERful,...
- Procedural or Declarative

## Decision point analysis

- Discovering e.g. decision trees on the XOR-splits, local decisions
- Decision Miner and offspring (Rozinat et al., de Leoni et al., Mannhardt et al.,...)
- Process mining with decision overlay

## Holistic approach

- BPMN + DMN (Batoulis et al.): still decision point analysis
- Artifact lifecycle discovery (Popova et al.)
- Treating data variables (for decisions) with the same priority as the event control flow



- ⇒ What is more challenging is to look for a model that does both at the same time: doing process and decision at same time.
- ⇒ Holistic approach



## CHAPTER 11: PRE-PROCESSING

⇒ We need learning techniques to deal with large amount of data, structured as well as unstructured data.

### 1. Data Generation

The World Wide Web contains about 170 terabytes of information on its surface; in volume this is seventeen times the size of the library of Congress print collections.

- Instant messaging generates five billion messages a day (750 GB) or 257 Terabytes a year
- Email generates about 400 000 terabytes of new information each year worldwide
- RFID
- QR code
- Bar code
- ...

⇒ Number of new applications that change companies and the way to work. Hence, there is a certain motivation to store data.

Initial reason	Potential
<ul style="list-style-type: none"> <li>- In telecommunication: billing</li> <li>- In supermarkets: inventory management</li> <li>- In banks: transactions</li> <li>- Productive sector: process control</li> </ul>	<ul style="list-style-type: none"> <li>- In telecommunication: fraud detection</li> <li>- In supermarkets: sales association</li> <li>- In banks: customer segmentation</li> <li>- Productive sector: preventive maintenance</li> </ul>

### 2. Pre-processing

Analytics steps

- Data exploration
- Pre-processing
- Technique selection
- Evaluation
- Interpretation of results

#### 2.1 Data exploration

In data mining projects, we start by exploring the data, we then transform it into the correct format.

Before going into data mining, we look at the data and compute basic statistics. We make use of some interesting exploration tools.

- Explore your dataset visually
  - Boxplot
  - Scatter plot
  - Histogram
  - Etc.
- Basic Statistics
  - Outlier detection
  - Correlation matrix

## 2.2 Pre-Processing

Before applying techniques, pre-processing is required. Remove some mistakes, deal with duplicates, what about outliers, etc. So, before going on we start **cleansing** the data.

- **Missing values:** ca can remove them or try to find out why they are missing. We can also replace these missing values by an average. We could delete rows, impute them, add separate columns, indicating why they are missing.
- **Duplicates:** if there is a high amount, how do we deal with this?
- **Identify outliers** (age, income, temperature, etc.). outliers are unexpected high values. Ex: age = 300. We often assume these are wrong values. However, some of them might be realistic values, hence we cannot always delete the data. Sometimes we can just try to correct the data, not leaving it like this.
- **Correct inconsistent data:** related to attribute combinations. Ex: a 5-year-old who is employed.

We then move on to a **transformation** step: from one category to another. We may also be interested in **normalization** and **standardization**.

- Attribute types:
  - Nominal vs. numerical
  - Regular vs. Label
  - Depends on your technique! Ex: a DT cannot cope well with numeric values.

- Normalization

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Standardization  
constrain feature to have **zero mean** and **stdev of one** ( $\sim N(0,1)$ )

$$x_{new} = \frac{x - \mu}{\sigma}$$

- Categorization (binning, grouping)
- Variable/Feature selection and transformation: which variables are useful to undertake a data mining process? We may have high number of variables and only choose some of them.
  - Transformation: reduce dimensionality, reduce problem to a simpler problem. Ex: PCQ combines variables into one variable. The principal component analysis: convert possibly correlated variables into a set of linearly uncorrelated variables called principal components
  - Feature extraction

### 2.3 Technique selection

Once the previous steps are completed, we do the **technique selection**, based on applications and a number of attributes. We also have to think about the visualization.

Keep characteristics of **techniques** in mind

- Numerical vs. nominal data
- Prone to overfitting?
- Black box: must the results be explainable?
- Visual: how are the results going to be visualized?
- Actionable: can we act on the newly gained insights?

→ Classification, regression, cross validation. Our basic model goes somewhat more complicated.

⇒ We can then move on to the **evaluation** phase and **visualization of results**.

### 3. Which features to use for predictive models?

**Feature selection**: how does feature selection happen? Many variables which are very much related. No need to take all of them, one might be enough. Often little hidden how they are related. Feature selection tries to figure out which variables are related and variables are wrapped up.

There is a certain relation between variables.

- Chi square test
- KS test
- ...

There are different approaches to feature selection

- **Filter:** “Usefulness” of features. Relation between features (independent variables) and goal variable (dependent variable)
  - Correlation
  - Chi-square
  - ANOVA, test Ks, ...

Important: filters are always independent of predictive models

### **CHI-SQUARE**

Given: two **categorical** variables

Hypothesis: these variables are **independent**

Independence: “knowing the value of one variable does not affect the probability distribution of the other variable’s values”

Contingency table (cross tabulation, cross tab): matrix with r rows & k columns, where, r = number of values of variable 1; k = number of values of variable 2.

Example:

Variable 1=Age, variable 2=gender

Degree of freedom:

$$df=(r-1)(k-1)$$

Basic idea:

Compare observed and Expected frequency

$H_0$ : variables are independent

r=2

	Gender		
Age	male	female	Total
< 30	60	50	110
>= 30	80	10	90
Total	140	60	200

k=2

Expected frequency in cell  $f_e$ :

	Gender		
Age	male	female	Total
< 30	60	50	110
>= 30	80	10	90
Total	140	60	200

$$f_e = (f_r * f_k) / n$$

with:

$f_r$  = total frequency in row r

$f_k$  = total frequency in column k

Example:  $r=k=1$ ;  $f_1=110$ ;  $f_1=140$ ;  $n=200$

$$f_e = (110 * 140) / 200 = 77$$

$H_0$ : Age and gender are independent

$H_1$ : Age and gender are not independent

$$df = 1 = (r-1) \cdot (k-1)$$

Critical value of chi-square distribution ( $df=1$ ,  $\alpha=0,01$ )=6,63 (see table)

$$\begin{aligned} \text{Chi-square} &= \sum \frac{(f_o - f_e)^2}{f_e} = \frac{(60-77)^2}{77} + \frac{(50-33)^2}{33} + \frac{(80-63)^2}{63} + \frac{(10-27)^2}{27} \\ &= 27.8 > 6.63 \Rightarrow \text{reject } H_0 \Rightarrow \text{Age and gender are not independent.} \end{aligned}$$

## Chi-square table

$\alpha = 0.01$

df=1

Degrees of Freedom	Probability of a larger value of $\chi^2$									
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01	0.001
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63	10.83
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21	13.82
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34	16.27
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28	18.48
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09	20.52
6	0.872	1.635	2.204	3.455	5.348	7.84	10.64	12.59	16.81	22.46
7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48	24.33
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09	26.19
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67	27.88
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21	29.59
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72	31.22
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22	32.91
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69	34.57
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14	36.19
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58	37.79
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00	39.36
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41	40.79
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80	42.16
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19	43.53
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57	44.91
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29	47.78
24	10.856	13.848	15.659	19.037	23.337	28.24	33.20	36.42	42.98	50.66
26	12.198	15.379	17.292	20.843	25.336	30.43	35.56	38.89	45.64	53.54
28	13.565	16.928	18.939	22.657	27.336	32.62	37.92	41.34	48.28	56.42
30	14.953	18.493	20.599	24.478	29.336	34.80	40.26	43.77	50.89	59.34
40	22.164	26.509	29.051	33.660	39.335	45.62	51.80	55.76	63.69	71.42
50	27.707	34.764	37.689	42.942	49.335	56.33	63.17	67.50	76.15	87.56
60	37.485	43.188	46.459	52.294	59.335	66.98	74.40	79.08	88.38	101.28

http

## ANOVA

Analysis of variance. The proportion of outcome variance explained by variable or group of variables

- **Wrapper**

We take features together; we then forward or backward eliminate. There are a number of techniques and metrics to do this.

Idea: which variables are interesting to take in the model and which are not. >> risk of overfitting + time computation.

→ Some methods combine wrapping and filtering

- **Forward selection:** start with small feature set and add iteratively "the next best feature" from a list of ranked features
- **Backward elimination:** start with complete feature set and eliminate iteratively "the least important feature" according to a feature ranking.

- **Stepwise selection:** combines forward selection and backward elimination
- **Metrics**
  - AIC (Akaike Information Criterion) minimize information loss
  - BIC (Bayes Information Criterion): starts from prior probabilities and maximizes posterior probability.

Disadvantages of filter approaches	Disadvantages of wrapper approaches
<ul style="list-style-type: none"> <li>- Each variable is assessed separately, not able to detect interactions between variables</li> <li>- Doesn't incorporate learning</li> <li>- Independent of modelling techniques</li> </ul>	<ul style="list-style-type: none"> <li>- Risk of overfitting</li> <li>- Computation time</li> <li>- Dependent on modelling technique</li> </ul>

- **Embedded methods:** combines feature selection with learning. Ex: LASSO (Least Absolute Shrinkage and Selection Operator): shrinkage method for continuous variable selection, minimizes penalized loss function.

We then move on the **feature extraction**. Basically, we try to reduce the dimensionality, to a simple distinction. Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases, leading to better human interpretations.

- **Principal Component Analysis (PCA):** statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the small of the number of original variables or the number of observations.

Measured variables:  $V_1, V_2$

Using  $Z_1, Z_2$  instead would explain "more of the variance" contained in the data.

This intuitive expression will be formalized.

Artificial data with  $n$  observations and  $m=3$  variables.

Let  $X = [x_1, x_2, x_3] \in \mathbb{R}^{n \times 3}$  be the data matrix.

	$x_1$	$x_2$	$x_3$
$x_1$	1.000	0.562	0.704
$x_2$	0.562	1.000	0.304
$x_3$	0.704	0.304	1.000
	$var(x_1) = 1$	$var(x_2) = 1$	$var(x_3) = 1$

Table: Correlation matrix

Variables are standardized => Variance = 1 !!

Problem (2) yields the following  $m$  solutions ordered by descending eigenvalue:

$u_1$	$u_2$	$u_3$
0.65	0.09	-0.76
0.51	0.80	0.33
0.57	-0.59	0.56
$\lambda_1 = 2.05$	$\lambda_2 = 0.72$	$\lambda_3 = 0.23$

Table: Solutions for problem (2)

We get the  $m = 3$  components  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3] \in \mathbb{R}^{n \times 3}$  by  $\mathbf{Z} = \mathbf{X}\mathbf{U}$  with  $\mathbf{X}$  the data matrix and  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_3]$ .

It can be shown:  $\lambda_i = \text{var}(\mathbf{z}_i)$ ;  $i = 1, \dots, m$ , i.e. concentration of variance in few components

- When is it appropriate to use PCA? Test if data has spherical structure as opposed to ellipsoidal. When the data is spherical, the dimension reduction is inappropriate.
- How should the data be scaled? Standardization makes numerical data from different scales comparable (Ex: sales vs. number of employees). However, we may not want to standardize data from a five-point Likert Scale.
- How many components should be retained? Scree plot: an elbow in the “component vs. eigenvalue” plot indicates an appropriate number of components.
- **Scree Plot:**
  - **Advantages:**
    - Easy to understand
    - Intuitive
  - **Disadvantages**
    - Not always clear where the elbow is
    - Cases with several elbows
    - Very subjective
- **Kaiser’s rule:** retain only components with eigenvalue exceeding unity
- **Explained variance:** retain a sufficient number of component in order to account for at least a pre-specified percentage of variance in each one of the original variables.
- **ICA:** Independent Component Analysis: used to separate statistically independent signals.  
Ex: Financial Time Series; Image Processing

#### 4. Related topics

- Data quality
- Data cleaning
- Data cleansing

##### 4.1 Data quality

Data quality is not specific to data mining but is important in the whole data mining world. High number of costs are related to data quality.

How do we obtain better information quality? **Completeness**, **accuracy**, how good is data? **Timeliness** (is data up to date?), to which period does the data refer?

- Data relevance: usefulness?
- Accuracy
  - Conform with standard or true value?
  - Potential causes: typo's, missing values, misspecification of attributes, ...
- Completeness
  - 3 perspectives of completeness: depth, breath, scope
  - does our data represent the true population, e.g. customers? (sample size)
  - data coverage (e.g. transaction date only captures data, and not time of day)
- Timeliness of data
  - Is your data up-to-date?
  - Is the worth of your data changing due to a lifecycle?
- Consistent
  - Is data in every database the same?
  - Ex: does a customer have the same address in every dataset?
- Coherent
  - Can you combine different data in a reliable way for different applications?
- Reliability
  - Do you have access to all data you need? Can you access these data in a fast and appropriate manner?
  - Redundancy and maintainability

>< Bad data quality

- **Missing values:** with missing values, it is interesting to know why some values are missing. We cannot just assume it is a coincidence and delete it. We have to do something: eliminate (rows or columns), replace (acquire missing values), impute, ... Replacing by the mean for instance is one technique.
- **Out-of-range data**
- **Data not up-to-date**
- **Redundant data**

Bad quality data may have some **consequences** ...

- Unsatisfied customers
- Increased operational costs
- Unsatisfied employees



#### 4.2 Imputation techniques

- Replace by mean
- Hot deck: method for handling missing data in which each missing value is replaced with an observed response from a “similar” unit.
- Regression: replace missing values of variable y using a regression model with x as independent variable(s)
- Regression with residuals: replace missing values based on a regression model using the respective residuals

Pros	Cons
<ul style="list-style-type: none"><li>- It is possible to apply data mining techniques that require complete data sets</li><li>- No loss of (correct) information</li></ul>	<ul style="list-style-type: none"><li>- Imputation introduces errors</li><li>- Is it worth the effort</li></ul>

#### 4.3 Outliers

⇒ Check whether attributes contain outliers

Reasons? Exceptional observations, wrong data entry.

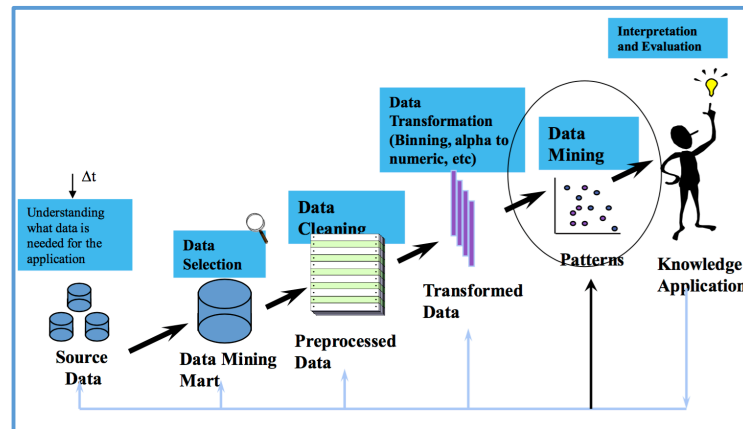
Should outliers be kept, omitted or replaced?

Techniques: **Winsorizing**: set all data to a specific percentile, e.g. set all data which fall out of 95<sup>th</sup> percentile to the 95<sup>th</sup> percentile.

## CHAPTER 12: EVALUATION

- ⇒ How well does the model fit with the data?
- ⇒ How well did we classify new observations according to the model?

In RapidMiner we will find this under validation performances.



Evaluation is part of the analytics steps we previously discussed.

### 1. Validation

Idea: we don't use all the data. We split our data set depending on our validation technique.

We first split our data set into training and test set. We derive a model for one part of the data and we test it on the other part of the data. We use this when we don't have too much data. This is a way to avoid bias due to particular division of data set.

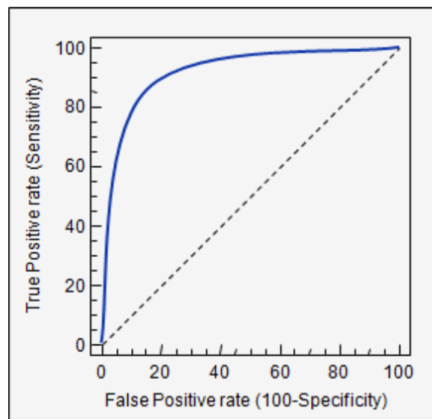
We often apply 70%-30%.

#### Cross validation:

- Multiple rounds
- K-fold: k subsets
- Leave 1 out: subsets of 1
- Use when
  - Small dataset
  - To avoid bias related to particular division in training and test set
  - To avoid overfitting and support generalization

### 2. Performance metrics

There exist some performance metrics to see whether instances were correctly classified or not: ROC curve, Confusion Matrix, etc.



- Correctly classified instances (TP + TN)
- Incorrectly classified instances (FP + FN)
- $\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$
- $\text{Sensitivity (recall)} = TP / (TP + FN)$
- $\text{Specificity} = TN / (FP + TN)$
- Confusion matrix
- ROC curve (and the corresponding area under curve (AUC))

Model	Actual		
		True	False
	True	True Positive (TP)	False Positive (FP)
	False	False Negatives (FN)	True Negatives (TN)

## CHAPTER 13: DATA VISUALIZATION

- ⇒ Have a look at enormous amount of data and try to have a first idea about clusters our trends.

When dealing with visualisation, there are some interesting attention points.

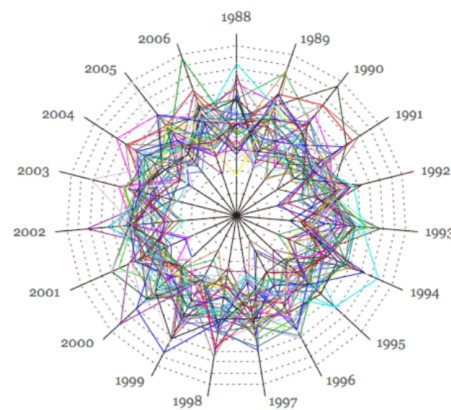
### 1. Introduction

**Challenge:** Definition of appropriate summaries and displaying these multivariate summaries in such a way that the complexity of the business process is captured in an understandable way for the observer.

**Ex:** A graphic is usually more instructive than a table of numbers.

**Bad example:** This visualization of winning lottery numbers by year is much too busy to show anything meaningful. It is fairly aesthetic, but doesn't convey meaningful data simply. Each coloured line represents a different number and the radii represent the number of draws of that number throughout. This might be more interesting if interactivity was added, or the creator emphasized what they were trying to bring out.

**Lotto numbers, like a star**



**Good Example:** This word cloud visualization representing the important topics collated from online submissions clearly shows what are the popular concerns. The most important topics stick out, but the viewer can still drill down and see the lesser issues easily. Words like "America" and "American" alone don't mean anything.



- ⇒ Bad visualization may be useless, nor reflecting interesting information.

Data visualization is of utmost importance in analytical Business Intelligence. It is always interesting to be able to present results to business people. Several steps of the KDD process require visual presentations:

- To get a first idea of the data at hand: are there any special cases here?

- Feature selection (correlation): what are interesting input variables and which of these are interesting?
- Transformation
- Presentation of patterns found

⇒ *“A picture is worth a thousand words (or numbers)”*

## 2. Data visualization principles

⇒ There are some things to care about!

- **Graphical Integrity:** visual representations of data must tell the truth.  
**Lie factor** can be calculated by dividing the size of the effect shown in the graphic by the size of the effect in the data.

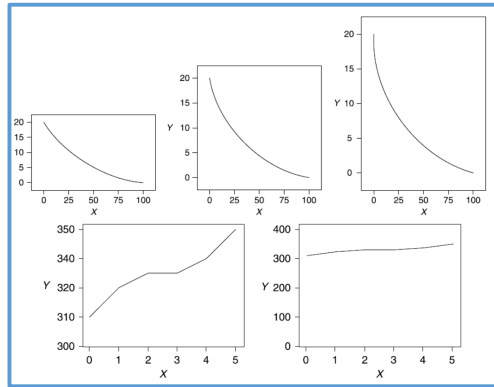
### 6 principles of Graphical Integrity

- 1) The **representation of numbers**, as physically measured on the surface of the graph itself, should be directly proportional to the numerical quantities represented. If something doubles in reality, the representation should not show that this is tripling. There should be a correspondence between the graph and the facts.
- 2) Clear, detailed and thorough **labelling** should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graph itself. Label important events in the data. To avoid confusions.
- 3) **Show data variation**, not design variation. We should not change the graph when drawing it.
- 4) In time-series displays of money, deflated and standardized units of monetary measurement are nearly always better than nominal units. What I am showing/measuring?
- 5) The number of information carrying (variable) dimensions depicted should not exceed the number of dimensions in the data.
- 6) Graphics must **not quote data out of context**.

Rmq: Changing the scale may change our interpretability. So, we have to be careful when reading graphs. On the picture below, some graphs show stable data, due to larger scales.

The three upper pictures are the same but the picture has been stretched. This gives the impression that there is a huge increase/decrease.

The same occurs if you play with the axes.



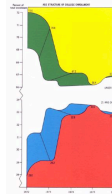
- **Data-ink:** how much useful information do we produce per unit of space? We cannot omit any part of the represented graphs. We need all information.

Data Ink = ink that represents data.

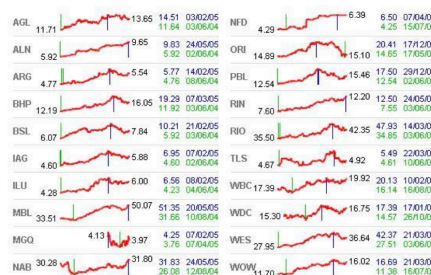
Data-ink ratio = 1 – (proportion of the graph that can be erased without loss of data-information).

Ex: electroencephalogram: a graph that records the electrical activity from the brain. Data-ink ratio = 1.

- **Chartjunk:** the excessive and unnecessary use of graphical effects in graphs (to attract attention). Shadows, colours, etc. are useless and do really kill the message.



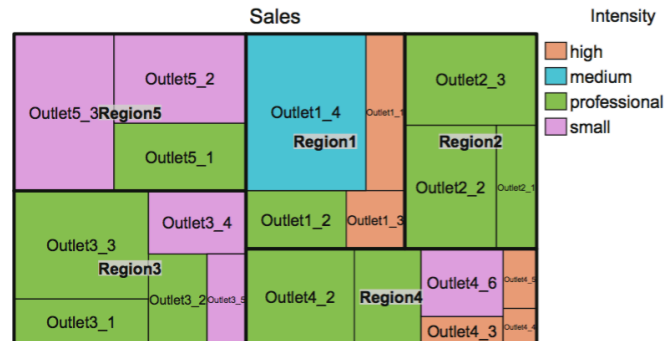
- **Data density:** the proportion of the total size of the graph that is dedicated displaying data. Most graphs can be shrunk way down without losing legibility or information.
- **Small Multiples:** series of the same small graph repeated in one visual. Ex: stock movements



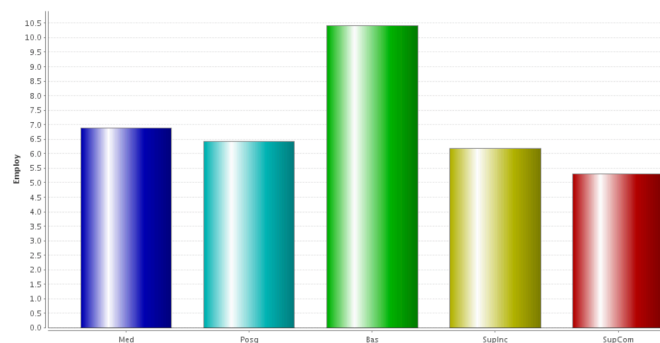
### 3. Best practices

⇒ Application of visualization techniques depends essentially on the type and number of variables and the complexity of the data structure.

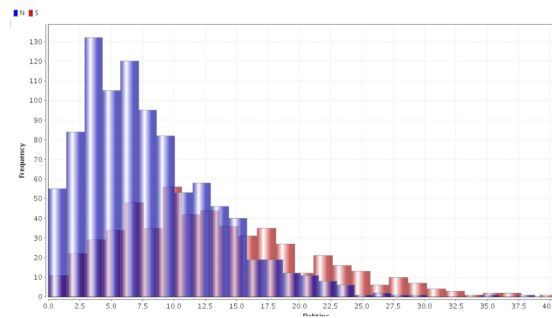
- **Tree Map:** In the case of a qualitative variable with values defining a nested hierarchy of groups, a tree map is frequently used. The hierarchy is shown by nested rectangles, and the size of the rectangles represents the value of interest for the quantitative variable. Colours can be used to display additional information.



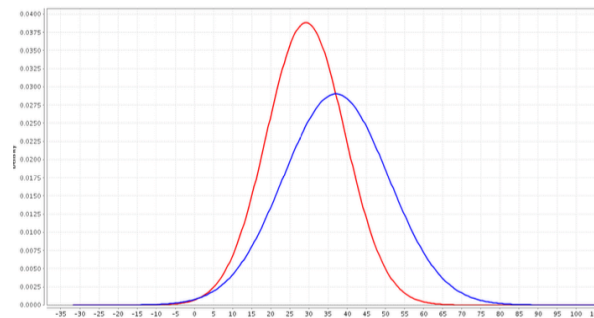
- **Bar Chart:** One variable can be visualized using a bar chart for absolute or relative frequencies or a pie chart for relative frequencies.



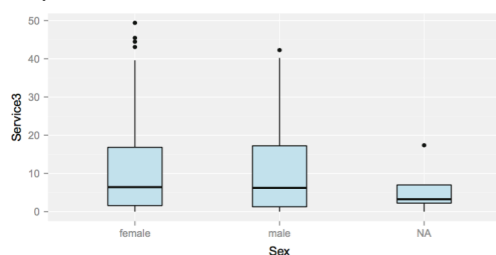
- **Histogram:** The definition of a histogram starts with the assignment of non-overlapping classes for the observations, so-called bins, which cover the entire range of the values. For each bin, the number of observations falling within the bin is counted. The height of the bars may be defined in different ways. The first one is defining the height of the bars by the absolute frequency (i.e., the counts), the second one is defining the height by relative frequency (i.e., percentage of observations within the class), and the third one is defining the heights in such a way that the area of the bars corresponds to the relative frequency of the bin. From a theoretical point of view, the best representation is the third one, which labels the height with the term density.



- **Distribution:** Density distribution of numerical variables

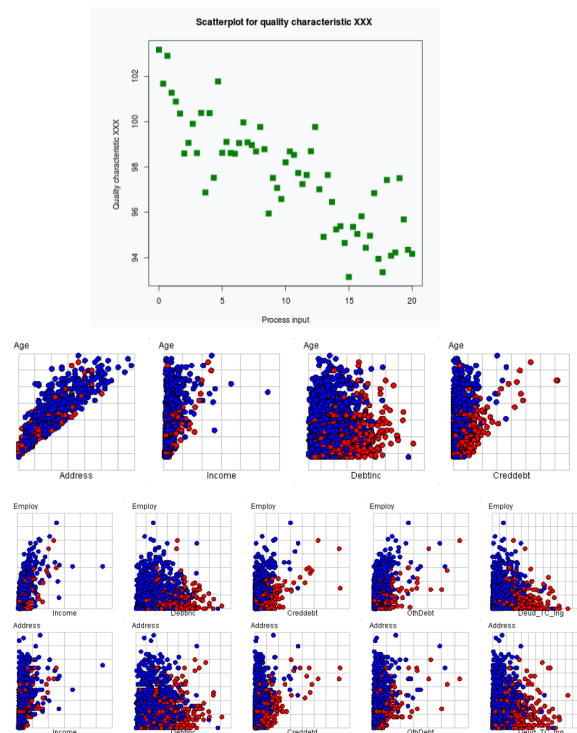


- **Box Plots:** A boxplot is a schematic representation using the quartiles. The 0:25 quantile and the 0:75 quantile define a box of the central 50 % of the observations, and the median is marked within the box. Furthermore, whiskers are defined on both ends of the boxes. They mark the area in which all the data should fall provided the data follow a normal distribution. Data outside the hinges are marked and considered as candidates for outliers, which deserve special consideration. Boxplots are a useful visualization if one wants to schematically compare the distribution of one variable in different groups.

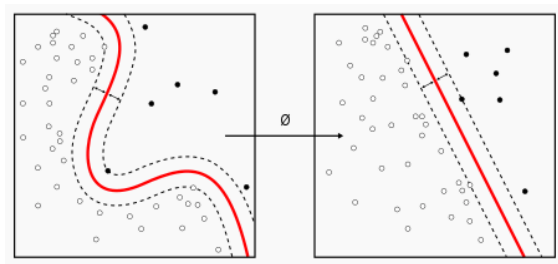


- **Scatterplots (2D or 3D):** A scatter plot is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. The data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.





- Self-Organizing maps (SOM):** A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction. Consists of neurons that captures information and trends.



## CHAPTER 14: UNCERTAINTY MODELING

### 1. Introduction

As we have previously seen, knowledge might come from different sources. Based on the knowledge we have, we will have to build certain models, before we even start working on it. Among this knowledge, there may be some uncertainty. There are several **causes** for uncertainty:

- Lack of information
- Abundance of information
- Conflicting evidence
- Ambiguity
- Measurement (errors)
- Belief
- ...

### 2. Approaches to deal with uncertainty

There exist a certain number of approaches to deal with uncertainty.

- **Probability theory**
- **Multi-valued logic**: means there is more than just “yes” and “no”. something can be true “to a certain extend”. Hence, there are all types of variations in answering a question.
- **Heisenberg principle**
- **Fuzzy set theory**
- **Evidential reasoning**: if I receive pieces of evidence, it increases my knowledge. Ex: piece of evidence for fraud.
- **Rough set theory**
- **Possibility theory**

#### 2.1 Probability theory

Probability theory is the branch of mathematics concerned with **probability**, the analysis of **random phenomena**. The central objects of probability theory are random variables, stochastic processes, and events: mathematical abstractions of non-deterministic events or measured quantities that may either be single occurrences or evolve over time in an apparently random fashion.

It is not possible to predict precisely results of random events. However, if a sequence of individual events, such as coin flipping or the roll of dice, is influenced by other factors, such as friction, it will exhibit certain patterns, which can be studied and predicted.

## 2.2 Lukasiewicz multi-valued logic

- Law of the excluded third: "For any proposition, either that proposition is true, or its negation is"
- 1917: Lukasiewicz introduced three-valued propositional calculus
- Lukasiewicz (1920): "On Three-valued logic".

In logic, a multi-valued logic is a propositional calculus in which there are more than two truth values. Traditionally, there were only two possible values (i.e., "true" and "false") for any proposition. Classical two-valued logic may be extended to n-valued logic for n greater than 2. Those most popular in the literature are three-valued, which accept the values "true", "false", and "unknown", the finite-valued (finitely-many valued) with more than three values, and the infinite-valued (infinitely-many valued), such as fuzzy logic and probability logic.

## 2.3 Heisenberg Principle

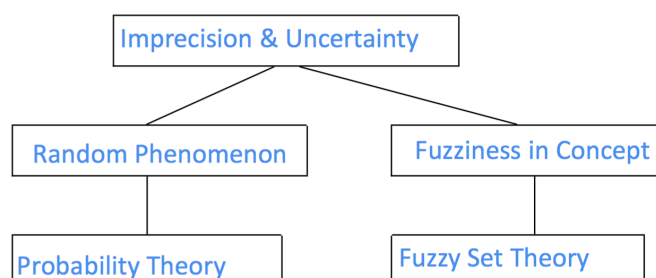
Heisenberg 1927: "The more precisely the position of some particle is determined, the less precisely its momentum can be known, and vice versa". One can never know with perfect accuracy the two variables that determine the movement of one of the smallest particles, its position and its velocity. It is impossible to determine accurately both the position and the direction and speed of a particle at the same instant.

⇒ Hence, we have to reason with this incomplete information.

## 2.4 Fuzzy set

- Zadeh (1965): "Fuzzy Sets"
- Bezdek (1973): "Fuzzy Clustering"
- Mamdani (1973): "Fuzzy control"
- 1975: First industrial application: Cement kiln, Denmark
- Fuzzy expert systems

⇒ In daily life, people have to think and reason with impression & uncertainty in many cases.



Examples of fuzzy (linguistic) terms are: most, more-or-less, much more than, tall, good, very good, very tall, approximately, about, nearly.

Fuzzy is totally different from probability theory. Fuzzy logic is not about a random phenomenon. We need a way to describe the concept which is fuzzy by nature. It is natural not random. Nevertheless, there will always be some imperfections which fuzzy tries to describe. Once we have fuzzy sets, we can use the set operators.

There exist different motivations for Fuzzy Sets

- **Principle of incompatibility:** as the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour decreases until a fixed threshold. Beyond this threshold, precision and significance become almost mutually exclusive characteristics.
- You can't determine the degree of vagueness of things until you try to make them precise.
- An educated mind is distinguished by the fact that it is content with that degree of accuracy which the nature of things permits, and by the fact that it does not seek exactness where only approximation is possible.



Let  $X$  be a classical set. A fuzzy set  $\tilde{A}$  is a set  $\{(x, \mu_{\tilde{A}}(x)); x \in X\}$ .

$\mu_{\tilde{A}}(x)$  is called the membership function of fuzzy set  $\tilde{A}$ .

**Example:**  
 $X$  = set of students at KU Leuven.  
 $\tilde{A}$  = fuzzy set of "tall students" at KU Leuven.

It is possible to operate on Fuzzy Sets.

Given fuzzy sets  $\tilde{A}$  and  $\tilde{B}$

The intersection  $\tilde{A} \cap \tilde{B}$  is a fuzzy set with membership function  $\mu_{\tilde{A} \cap \tilde{B}}(x) := \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)) \quad x \in X$

The union  $\tilde{A} \cup \tilde{B}$  is a fuzzy set with membership function  $\mu_{\tilde{A} \cup \tilde{B}}(x) := \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)) \quad x \in X$

The complement of a fuzzy set  $\tilde{A}$  is a fuzzy set with membership function  $\mu_{\tilde{A}^c}(x) := 1 - \mu_{\tilde{A}}(x) \quad x \in X$

### Fuzzy Database retrieval

Query: 'List the YOUNG salesmen who have a GOOD selling record for HOUSEHOLD GOODS in the NORTH of England'

In a comprehensive manner for classical systems: 'List salesmen under 25 years old who have sold more than £20,000 of goods in the categories ... to shops in the regions ...'

→ A man of 26 years old selling £70,000 ?

- **Fuzzy Data Representation:** “What and how is fuzziness introduced in the model?”
- **Fuzzy Data Manipulation:** “How can databases be queried in fuzzy terms?”
- **Fuzzy Database Design:** “How can we properly design a fuzzy RDB?”

Some distinction is required between probability theory and fuzzy set theory

- **Probability theory:** partial information about a future (precisely defined) event
- **Fuzzy set theory:** complete information about an imprecisely (maybe subjectively) described event.

### Linguistic Variable

- A linguistic variable is a quintuple  $(x, T(x), X, G, M)$  where:
  - $x$  is the name of the variable
  - $T(x)$  is the set of linguistic values (or primary terms)
  - $X$  is the universe of discourse
  - $G$  is a syntactic rule that generates the linguistic values
  - $M$  is a semantic rule which provides meanings for the linguistic values
- “Age” as linguistic variable  $(x, T(x), X, G, M)$ :
  - $x$ : name of the variable ( $x = \text{“Age”}$ )
  - $T(x)$ : set of linguistic values or primary terms ( $T(x) = \text{“young”, “medium-aged”, “old”}$ )
  - $X$ : universe of discourse ( $X = [0, 100]$ )
  - $G$ : syntactic rule that generates the linguistic values (operators that modify basic terms such as “not”, “very ...”, ...)
  - $M$ : semantic rule which provides meanings for the linguistic values (fuzzy sets for each term from  $T(x)$ )

It is possible to modify these linguistic variables by applying a so-called “linguistic modifier”. Take for instance linguistic terms as values: young, tall, more-or-less, very, most, etc.

Example of linguistic modifiers are:

- concentration  $\text{con}(A) = \text{very } A$

$$\mu_{\text{con}(A)}(x) = (\mu_A(x))^2$$

- dilatation  $\text{dil}(A) = \text{more-or-less } A$

$$\mu_{\text{dil}(A)}(x) = (\mu_A(x))^{1/2}$$

- A = set of "recent" years:  
 $A = \{(2002,1), (2001,0.8), (2000,0.6), (1999,0.4)\}$
- "Very recent":  
 $\text{con}(A) = \{(2002,1), (2001,0.64), (2000,0.36), (1999,0.16)\}$
- "More-or-less recent":  
 $\text{dil}(A) = \{(2002,1), (2001,0.89), (2000,0.77), (1999,0.6)\}$

## Fuzzy Logic

The Fuzzy Logic is different to Fuzzy Set Theory (= generalization of classical sets). The Fuzzy Logic is the **generalization** of classical Boolean logic.

A	B	A and B	A or B	A xor B
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
1	1	1	1	0

Boolean logic

A, B fuzzy statements defined by fuzzy sets  
 A "and" B (minimum, product)  
 A "or" B (maximum, algebraic sum)

Fuzzy logic (1965)

Fuzzy rules can be represented by an "if-then" format. If x is A then y is B (where A & B are linguistic values defined by fuzzy sets on universes of discourse X & Y).

- "x is A" is called the **antecedent** or premise
- "y is B" is called the **consequence** or conclusion

### Examples:

- If pressure is high, then volume is small.
- If the road is slippery, then driving is dangerous.
- If a tomato is red, then it is ripe.
- If the speed is high, then apply the brake a little.
- A fuzzy if-then rule can be viewed as a fuzzy relation R.

Fuzzy reasoning is also known as approximate reasoning. It is an inference procedure that derives conclusions from a set of fuzzy if-then-rules and known facts.

**Given A,  $A \Rightarrow B$ , infer B (here still without fuzzy)**

A = "today is sunny"

$A \Rightarrow B$ : day = sunny then sky = blue

infer: "sky is blue"

Matching problem: try to see if the input data matches the rules to a **certain extend**.

"Equality" of two fuzzy sets (or poss. distr.):

$$E(A, B) \in [0,1]$$

(how close are they?)

Various definitions

e.g., via height, distance, etc.

Example:

$$\begin{aligned} E(A,B) &= \text{height}(A \cap B) \\ &= \sup_x \min(m_A(x), m_B(x)) \end{aligned}$$

### Generalized Modus Ponens

classical:

$$a, a \rightarrow b \Rightarrow b$$

generalized (variables X,Y on U,V resp.):

rule: if X is A then Y is B

fact: X is A'

---

conclusion: Y is B'

B' can be derived as follows:  $\forall v \in V,$

$$m_{B'}(v) = \sup_{u \in U} \min(m_A(u), I_{GB}(m_A(u), m_B(v)))$$

#### Definition of fuzzy reasoning

Let A, A' and B be fuzzy sets of X, X, and Y, respectively.

Assume that the fuzzy implication  $A \Rightarrow B$  is expressed as a fuzzy relation R on  $X \times Y$ .

Then the fuzzy set B induced by "x is A'" and the fuzzy rule "if x is A then y is B" is defined by:

$$\mu_{B'}(y) = \max_x \min[\mu_{A'}(x), \mu_R(x, y)]$$

#### Single rule with single antecedent

Rule : if x is A then y is B

Fact: x is A'

---

Conclusion: y is B'

rule: if Quantity is HIGH then Discount is BIG

fact: the quantity is more-or-less high

Discount ?

given: A = "HIGH", B = "BIG", and

A' = "more-or-less HIGH"

we need to derive B'.

## 2.5 Theory of Evidence

Theory of evidence means "I receive information" even if my information is not perfect, I can still use it. I deal with information which is not perfect. Look at subsets of possibilities "power set". In probability theory we assign probabilities to events.

The theory of belief functions, also referred to as evidence theory is a general framework for reasoning with uncertainty, with understood connections to other frameworks such as probability, possibility and imprecise probability theories. The theory allows one to combine evidence from different sources and arrive at a degree of belief (represented by a mathematical object called belief function) that takes into account all the available evidence.

In this formalism a degree of belief (also referred to as a mass) is represented as a belief function rather than a Bayesian probability distribution. Probability values are assigned to sets of possibilities rather than single events: their appeal rests on the fact they naturally encode evidence in favour of propositions.

Dempster–Shafer theory assigns its masses to all of the non-empty subsets of the propositions that compose a system—in set-theoretic terms, the power set of the propositions. For instance, assume a situation where there are two related questions, or propositions, in a system. In this system, any belief function assigns mass to the first proposition, the second, both or neither.

Let  $X$  be a set of elements and  $\Omega$  be its power set (i.e. set of all subsets of  $X$ ).

The function  $m: \Omega \rightarrow [0,1]$  is a basic belief assignment (bba), iff

$m(\emptyset) = 0$  and

$$\sum_{A \in \Omega} m(A) = 1$$

$m(A)$  is also called mass of  $A$ .

Shafer's formalism starts from a set of possibilities under consideration, for instance numerical values of a variable, or pairs of linguistic variables like "date and place of origin of a relic" (asking whether it is antique or a recent fake).

Shafer's framework allows for belief about such propositions to be represented as intervals, bounded by two values, belief (or support) and plausibility: **belief**  $\leq$  **plausibility**.

In a first step, subjective probabilities (masses) are assigned to all subsets of the frame; usually, only a restricted number of sets will have non-zero mass. **Belief** in a hypothesis is constituted by the sum of the masses of all sets enclosed by it. It is the amount of belief that directly supports a given hypothesis or a more specific one, forming a lower bound. Belief (usually denoted  $Bel$ ) measures the **strength of the evidence** in favour of a proposition  $p$ . It ranges from 0 (indicating no evidence) to 1 (denoting certainty).

**Plausibility** is 1 minus the sum of the masses of all sets whose intersection with the hypothesis is empty. Or, it can be obtained as the sum of the masses of all sets whose intersection with the hypothesis is not empty. It is an upper bound on the possibility that the hypothesis could be true, i.e. it "could possibly be the true state of the system" up to that value, because there is only so much evidence that contradicts that hypothesis.

Plausibility (denoted by  $Pl$ ) is defined to be  $Pl(p) = 1 - Bel(\sim p)$ . It also ranges from 0 to 1 and measures the extent to which evidence in favor of  $\sim p$  leaves room for belief in  $p$ .



$$\begin{aligned}
 \text{Belief: } bel(A) &= \sum_{B \subseteq A} m(B) \\
 \text{Plausibility: } pl(A) &= \sum_{B \cap A \neq \emptyset} m(B) \\
 bel(A) &\leq P(A) \leq pl(A) \\
 pl(A) &= 1 - bel(\bar{A})
 \end{aligned}$$

Ex: Suppose that it is expected that a corporate take-over will happen soon. There are three possible raiders called Raider A, Raider B and Raider C.

$$\Rightarrow X = \{\text{Raider A, Raider B, Raider C}\}$$

$$\Rightarrow \Omega = \{\{A, B, C\}, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{\}\}$$

Suppose it is not possible that two raiders join forces for the take-over. Furthermore, one of these three must take over. This means that the possibilities are exhaustive and mutually exclusive.

It is possible to attribute masses of belief to the subsets of X.

- $m(\{A\})$  = the belief that Raider A will take over = 0.1
- $m(\{B\})$  = the belief that Raider B will take over = 0.2
- $m(\{C\})$  = the belief that Raider C will take over = 0.4
- $m(\{A, C\})$  = the belief that Raider A or B will take over = 0.15
- $m(\{B, C\})$  = the belief that B or C will take over = 0.05

All the other subsets get ZERO BELIEF because these situations are considered to be impossible. Note that in this theory the unawarded portion of belief goes automatically to X. This implies that  $m(\{A, B, C\}) = 1 - (0.1 + 0.2 + 0.4 + 0.15 + 0.05) = 0.10$

$$\begin{aligned}
 bel(\{A\}) &= m(\{A\}) \\
 &= 0.1
 \end{aligned}$$

The belief function = measure of belief for a singleton

$$\begin{aligned}
 bel(\{A, C\}) &= m(\{A, C\}) + m(\{A\}) + m(\{C\}) \\
 &= 0.15 + 0.1 + 0.4 = 0.65
 \end{aligned}$$

The belief that A or B will take over is 0.65

$$\text{In classical theory : } P(A \cup B) = P(A) + P(B) = 0.5$$

A and B are supposed to be independent.

$$\text{bel}(\{A\}) = 0.1$$

The hypothesis that A is the raider has belief = 0.1

$$\begin{aligned}\text{pl}(\{A\}) &= 1 - \text{bel}(\sim\{A\}) \\ &= 1 - \sum_{X \subseteq \{B,C\}} m(X) \\ &= 1 - (m(\{B\}) + m(\{C\}) + m(\{B,C\})) \\ &= 1 - (0.2 + 0.4 + 0.05) \\ &= 0.35\end{aligned}$$

The plausibility that A will take over is 0.35

**[bel(H), pl(H)] : measure of uncertainty**

Applying this to the example gives:

0 0.1      0.35      1  
[---|\*\*\*\*\*|-----]

Given the available material it can be said that A will take over with an uncertainty of [0.1 0.35].

## 2.6 Certainty Factor Theory

- Representing uncertain evidence
- Representing uncertain rules
- Combining evidence from multiple sources.

Ex: IF A and B THEN X cf 0.8  
IF C THEN X cf 0.7

What is the certainty of X?

$$-1 \leq \text{CF} \leq 1$$

$$-100 \leq \text{CF} \leq 100$$

**Used in**

- rule conclusions, values of variables in premise, answers to user queries

A **certainty factor (CF)** is a numerical value that expresses a degree of subjective belief that a particular item is true. The item may be a fact or a rule.

It is a numerical value that expresses the extent to which, based on a given set of evidence, we should accept a given conclusion. A Certainty Factor or CF with a value of 1 indicates total belief, whereas a CF with a value of -1 indicates total disbelief.

In a system that uses CFs, the rules must be so structured that any given rule either adds to belief in a given conclusion or adds to disbelief.

Rule: IF E THEN H CF(Rule)

$$CF(H,E) = CF(E) * CF(Rule)$$

Example: IF econ-two-years = strong

THEN likelihood-of-inflation = strong CF 40

Given econ-two-years = strong with CF = 70

$$CF(\text{likelihood-of-inflation} = \text{strong}) = (40 * 70) / 100 = 28$$

#### Conjunctive:

IF E1 and E2 and ... and En

THEN H CF(Rule)

$$CF(H, E1 \text{ and } E2 \text{ and } \dots \text{ and } E_n) = \min\{CF(E_i)\} * CF(Rule)$$

#### Disjunctive

IF E1 or E2 or ... or En

THEN H CF(Rule)

$$CF(H, E1 \text{ or } E2 \text{ or } \dots \text{ or } E_n) = \max\{CF(E_i)\} * CF(Rule)$$

#### Premise with AND (conjunctive)

Example: IF economy-two-years = strong

AND availability-of-investment-capital = low

THEN likelihood-of-inflation = strong

$$CF \text{ of condition} = \min(cf1, cf2)$$

#### Premise with OR (disjunctive)

Example: IF economy-two-years = poor

OR unemployment-outlook = poor

THEN economic-outlook = poor

$$CF \text{ of condition} = \max(cf1, cf2)$$

#### Certainty propagation in similarly concluded rules

R1: IF E1 THEN H CF1

R2: IF E2 THEN H CF2

(supporting evidence increases our belief)

$$Cf_{\text{combine}}(CF1, CF2)$$

$$= CF1 + CF2(1 - CF1), \text{ when both } > 0$$

$$= (CF1 + CF2) / (1 - \min(|CF1|, |CF2|)), \text{ when one } < 0$$

$$= CF1 + CF2(1 + CF1), \text{ when both } < 0.$$

#### Premise with both AND and OR

Example: IF has-credit-card = yes (cf = 80)

OR cash = ok (cf = 90)

AND payments = ok (cf = 85)

THEN approval = ok

(has-credit-card = yes AND payments = ok)

$$[\min(80, 85)]$$

OR

(cash = ok AND payments = ok)

$$[\min(90, 85)]$$

$$CF = 80 + 85 - (80 * 85) / 100 = 97$$

→ An example is provided in the slides

## 2.7 Possibility Theory

Possibility theory is a mathematical theory for dealing with certain types of uncertainty and is an alternative to probability theory.

**“Hans ate u eggs for breakfast.”**

The values of  $\pi_x(u)$  and  $P_x(u)$  might be as shown in the following table:

**Possibility:**

**Probability:**

$u$	1	2	3	4	5	6	7	8
$\pi_x(u)$	1	1	1	1	0.8	0.6	0.4	0.2
$P_x(u)$	0.1	0.8	0.1	0	0	0	0	0

$\Pi(x)$ : The possibility of  $x$  in a possibility distribution  $\Pi_A$  reflects the possibility that variable  $X$  takes the value  $x$

## CHAPTER 15: KNOWLEDGE DISCOVER IN DATA

Once again we are studying knowledge discovery and analytics. We hereby deal with unsupervised learning. Which means that, based on a bunch of data, we look at interesting patterns over time. We don't have a labelled data set. We try to find human interpretable patterns.

As a reminder, we studied two different data mining tasks and techniques.

- **Predictive:** classification and regression
- **Descriptive:** you don't have a labelled data set unsurprised. The main purpose here is to find human-interpretable patterns that describe the data.  
Ex: clustering, associations, sequences.

### 1. Association rule mining

Idea: Association analysis works on event sets. Typically, these event sets are produced by transactions such as purchase transactions in an online store or prescription transactions in a hospital. Of particular interest are the items that are associated with the transactions.

Goal: The goal is to find association rules in the form of " $A \rightarrow B$ ," i.e., the occurrence of A implies the occurrence of B. We denote A as the antecedent and B as the consequent of the rule.

Afterwards, these association patterns are formulated as rules which are interest from a practical point of view.

#### Main Steps in Association Analysis

1. *Finding large item sets:* A minimal support is defined. Then, the task is to find all item sets for which their support is above the minimal support, so-called *large item sets*. The search might be confined by syntactical constraints. Finding large item sets can quickly become expensive as, in principle, the support of all possible combinations of item sets in the transaction set has to be determined. Optimization techniques, such as pruning, can be applied.
2. *Discover rules within large item sets based on confidence and syntactical constraints.* Given a set of large item sets, all possible rules with elements of this large item set are determined. We choose all those rules that exceed a given confidence. The result is a set of rules with a minimal support and minimal confidence that possibly respect certain syntactical constraints.

#### 1.1 Basic semantics

Association rule learning is a method for discovering interesting relations between variables. It is intended to identify strong rules discovered in databases using some measures of interestingness.

For example, the rule {onions, tomatoes, ketchup}  $\rightarrow$  {burger meat} found in the sales data of a supermarket would indicate that if a customer buys onions, tomatoes and ketchup together, they are likely to also buy hamburger meat, which can be used e.g. for promotional pricing or product placements.

The association rule mining has many application domains: application areas in market basket analysis, web usage mining, intrusion detection, production and manufacturing.

As opposed to sequence pattern mining, association rule learning typically does not consider the order of items either within a transaction (sequence mining does)

⇒ Pioneering technique: **Apriori algorithm** – Rakesh Agrawal

Association rule mining gained into importance with the Apriori algorithm. Variables are products in transactions. The market basket analysis tries to find out whether products are often bought together. This gives us some rules. We then have to discover which rules are more interesting.

Basically we have a database of transactions consisting of a series of products in whatever order. We look for association rules, saying that some products lead to other products. There is no time damage in association rules, no sequence is involved here, just observations of items in a transaction.

- D: database of transactions  $t_p$  (tuples)
- Each transaction  $t_p$  consists of a transaction ID and a set of items, selected from all possible items I.

**An association rule is an implication of the form:**  
 $X \Rightarrow Y$  where  $X \subset I, Y \subset I$  and  $X \cap Y = \emptyset$

Ex: if a customer buys spaghetti, then the customer buys red wine in 70% of the cases.

But note that association rules are not only used in shopping baskets.  
 E.g. {browser="firefox", pages\_visited < 10}  $\Rightarrow$  {sale\_made="no"}

## 1.2 Mining association rules

What constitutes a good rule? To select rules from the set of all possible rules, we usually put constraints on various measures of interest. The best known and most used constraints are the minimum thresholds on **support** and **confidence**.

- **Support:** percentage of total transactions in the database that comprise the itemset. The rule  $X \Rightarrow Y$  has support s if s% of the transactions in D contain  $X \cup Y$

$$\text{Sup}(X \Rightarrow Y) = \frac{\text{number transactions supporting } X \cup Y}{\text{total number of transactions}} = p(X \cup Y)$$

- **Confidence:** the rule  $X \Rightarrow Y$  has confidence c if c% of the transactions in D that contain X also contain Y

$$\text{Conf}(X \Rightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)} = \frac{p(X \cup Y)}{p(X)} = p(Y|X)$$

- **Lift** is the degree to which both occurrences are dependent on one another.

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X) \times \text{support}(Y)}$$

Lift tries to find out whether there is independence between the products.

- Lift > 1: degree of dependence of the two occurrences. These are not unrelated
- Lift < 1: Probabilities of both occurrences are independent.

### 1.3 A-Priori Algorithm

The Apriori algorithm is a two-step process

- Step 1: Identification of all (large) itemsets having support above minsup, i.e. “frequent” itemsets;
- Step 2: Discovery of all derived association rules having confidence above minconf.

Finding all frequent itemsets by searching all possible itemsets would be difficult: the power set over I has size  $2^n - 1$  (excluding the empty set which is not a valid itemset); 100 items  $\Rightarrow$  1.27 billion possibilities

An efficient search is possible using the downward-closure property of support which guarantees that for a frequent itemset, all its subsets must also be frequent and thus for an infrequent itemset, all its supersets must also be infrequent.

#### Basic notion:

- Every subset of a large itemset must be a large itemset (Apriori property). For infrequent itemsets, all subsets also need to be infrequent.
- **Join step:** so candidate itemsets having k items can be found by joining large itemsets having (k-1) items
- **Prune step:** and deleting those sets that contain any subset that is not large (not enough support)
- This results in a much smaller number of candidate itemsets
- Developed by Agrawal et al.

```

1)  $L_1 = \{\text{Large 1-itemsets}\};$ 
2) for ( $k = 2; L_{k-1} \neq 0; k++$ ) do begin
3)    $C_k = \text{apriori-gen}(L_{k-1});$  // New candidates
4)   forall transaction  $t \in D$  do begin
5)      $C_t = \text{subset}(C_k, t);$  //Candidate contained in t
6)     forall candidates  $c \in C_t$  do
7)        $c.\text{count}++;$ 
8)   end
9)    $L_k = \{c \in C_k \mid c.\text{count} \geq \text{min sup}\}$ 
10) end
11)  $\text{Answer} = \bigcup_k L_k$ 

```

#### Join Step

```

insert into  $C_k$ 
select  $p.\text{item}_1, p.\text{item}_2, \dots, p.\text{item}_{k-1}, q.\text{item}_{k-1}$ 
from  $L_{k-1} p, L_{k-1} q$ 
where  $p.\text{item}_1 = q.\text{item}_1, p.\text{item}_2 = q.\text{item}_2, \dots, p.\text{item}_{k-2} = q.\text{item}_{k-2},$ 
 $p.\text{item}_{k-1} < q.\text{item}_{k-1};$ 

```

#### Pruning Step

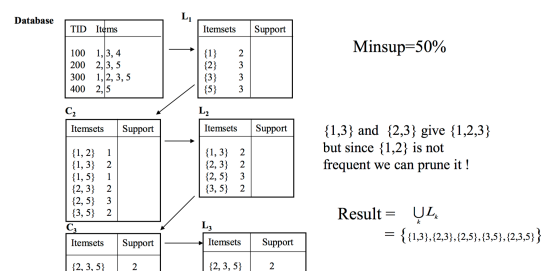
```

forall itemset  $c \in C_k$  do
  forall  $(k-1)$ -subsets  $s$  of  $c$  do
    if ( $s \notin L_{k-1}$ ) then
      delete  $c$  from  $C_k$ 

```

Ex: we start with a number of transactions; we always need frequent itemsets occurring in transactions. Take for instance product 4, it only occurs in one transaction, so we can drop this product for ever.

Drop all occurrence which do not meet 50%. Now we have 4 tuples, we start combining once again.



Once the frequent itemsets have been obtained, the association rules can be generated as follows:

- For each frequent itemset  $I$ , generate all nonempty subsets of  $I$ .
- For every nonempty subset  $s$  of  $I$ , output the rule  $s \Rightarrow I - s$  if the confidence  $>$  minconf.

Ex: for frequent itemset {cheese, wine, spaghetti} generate:

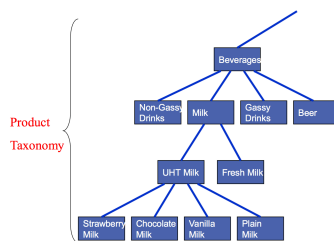


```

{cheese, wine} => {spaghetti},
{cheese, spaghetti} => {wine},
{wine, spaghetti} => {cheese},
{cheese} => {wine, spaghetti},
{spaghetti} => {cheese, wine},
{wine} => {spaghetti, cheese}

```

There is a large efficiency gain thanks to this algorithm. Once frequent itemsets have been found, association rules can be generated with sufficient support and sufficient confidence.



It is possible to extend this algorithm and mine association rules at different concept levels.

Ex: chocolate and Milk → Beer.

Add all higher-level items when lower-level appears in transaction. For instance, one specific type of beer leads to purchasing a certain product.

## 2. Sequence pattern mining

⇒ Detect temporal patterns between items

It would be interesting to look at the sequence of things, the order of bought items. We are now looking for temporal patterns. We want to form combinations of for instance 2 frequent sequences.

“If I buy this product today, will I buy that product tomorrow?”.

Sequence mining has many application fields:

- Store context
- Web use
- DNA

We want to look at certain customers. We need some ways to identify the customer. We need the information about who you are.

- Customer buys product X, then product Y, then product Z, ...
- 60% of clients who placed an online order in company/products/product1.html, also placed an online order in /company1/products/product4 within 15 days.

A commonly used **technique** is the modified Apriori.

- **Sets:** unordered, each element appears at most once: every transaction is a set of items
- **Sequences:** ordered, every transaction is a sequence of items.

id	milk	bread	...	id	Itemset	id	sequence
101	1	1	...	101	{milk, bread, beer}	101	<{milk}, {bread, beer}>
102	0	1	...	102	{bread, beer, cheese, wine}	102	<{bread, beer}, {cheese, wine}>
103	1	1	...	103	{milk, bread, cheese, spaghetti}	103	<{milk, bread, cheese}, {spaghetti}>
104	0	0	...	104	{cheese, wine, spaghetti}	104	<{cheese, wine, spaghetti}>
105	1	1	...	105	{milk, bread, cheese, wine, spaghetti}	105	<{milk}, {bread}, {cheese, wine}, {spaghetti}>

A **sequence** is an ordered list of sets  $S = \langle s_1, s_2, \dots \rangle$

- A sequence  $A$  is a subsequence of another sequence  $B$  if  $|B| > |A|$  and there exists a series of indexes  $i_1 < i_2 < i_3 < \dots$  so that  $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \dots$  For example,  $\langle \{2\}, \{3, 5\} \rangle$  is a subsequence of  $\langle \{2, 4\}, \{3\}, \{3, 5, 8\}, \{8\} \rangle$
- $\langle \{2\}, \{3, 5\} \rangle$  is not a subsequence of  $\langle \{2, 4\}, \{3\}, \{5\}, \{2\} \rangle$
- Based on this, support and confidence can be calculated

id	sequence	<{milk}, {bread}> subsequence?
101	<{milk}, {bread, beer}>	yes
102	<{bread, beer}, {cheese, wine}>	no
103	<{milk, bread, cheese}, {spaghetti}>	no
104	<{cheese, wine, spaghetti}>	no
105	<{milk}, {bread}, {cheese, wine}, {spaghetti}>	yes

Mining of frequent sequences: algorithm very similar to apriori

Start with the set of frequent 1-sequences

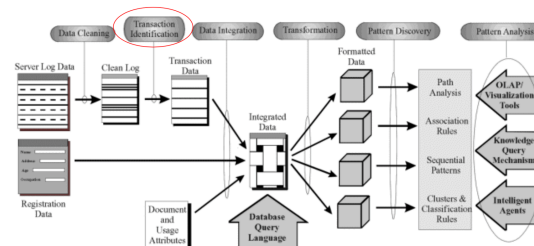
(Normal apriori started with frequent sets with size 1, e.g. {milk})

But: expansion (candidate generation) done differently

E.g. in normal apriori, {A, B} and {A, C} would both be expanded into the same set {A, B, C}

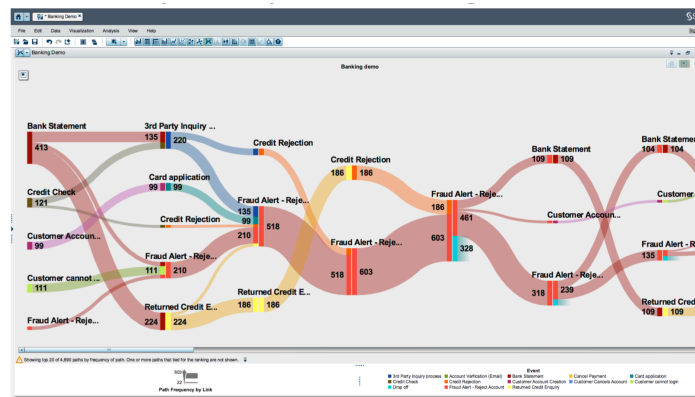
For sequences, suppose we have  $\langle \{A\}, \{B\} \rangle$  and  $\langle \{A\}, \{C\} \rangle$ , then these are now expanded into  $\langle \{A\}, \{B\}, \{C\} \rangle$  and  $\langle \{A\}, \{C\}, \{B\} \rangle$  and  $\langle \{A\}, \{B, C\} \rangle$

Pruning k-sequences with infrequent k-1 subsequences, only continue with support higher than threshold



This figure provides the same idea of finding association rules. It starts from the server log data. Identify transactions. "if customer does this and this on my website then probably the next step will be this". Combining this with the buying behaviour could lead to a supervised learning.

⇒ How do clients use the same web page?



The above screenshot gives an idea of how customers moved on the webpage. It gives indications about frequent pages.

⇒ ***An example of fraud analysis is provided in the slides***

## CHAPTER 16: KNOWLEDGE MANAGEMENT

### 1. Introduction to Knowledge Management

Knowledge management, yet another buzz word? Key to economic survival.

- Sharing knowledge: knowledge is power vs. shared knowledge is power
- I know → We know
- Transfer of knowledge in people's head to something lasting
  - Globalization
  - Less employee retention
  - Many people retiring

Knowledge might come from

- Knowledge discovery
- Knowledge engineered rules and decisions  
= knowledge we have written down
- Human/hidden knowledge
- External knowledge

What is knowledge? *"The message is that there are no knowns. There are things we know that we know. There are known unknowns. That is to say there are things that we now know we don't know. But there are also unknown unknowns. There are things we don't know we don't know."*

How immense is the problem to gather the knowledge, analyse it and act on it? Make sure all knowledge sources are well managed. Managing knowledge in a company is a big issue.

KM is a broad area, ranging from Human Resource perspective, trying to find the right people organizing discovered knowledge. How can we get it distributed to make sure everyone can reuse this knowledge? Next, in a company, it is important to capture knowledge before people retire or leave the company.

⇒ Hard to precisely find what is really knowledge. This already makes it complicated to integrate knowledge.

### 2. DIKW

- **Data:** raw observed facts, no context. Data is not information, information is not knowledge, knowledge is not wisdom. Wisdom is not truth.  
Ex: 100, 3%
- **Information:** data in context and with meaning. What, who, when, where  
Ex: 100 euros on bank account; 3% interest
- **Knowledge:** information interpreted with own insights and experiences that can lead to concrete actions. How?

Ex: At the end of the year, I will receive 3 euros, depositing more money will yield more interests, taking money, less interests

- **Wisdom:** Wisdom embodies more of an understanding of fundamental principles, foreseeing consequences. Why.

Ex: Action which produces a result which encourages more of the same action produces an emergent characteristic called growth. Nothing grows forever for sooner or later growth runs into limits.



Knowledge is not just data. It comes from different sources. We may have to deal with group knowledge, tacit knowledge, explicit, implicit knowledge, etc.

- Explicit knowledge
  - Has been articulated, codified, made public. It has been codified in for instance documents, reports, charts, papers, procedures, ...
  - Ex: set of rules for calculating taxes
- Tacit knowledge: exists within a person's mind, is private and unique. Tacit knowledge is often unconscious, based on personal experiences, individual learning, special know how. However, tacit knowledge is difficult to extract and codify.
  - Ex: experience of investment manager in trading stocks

*"Knowledge management is the process of creating value from an organization's intangible assets"*

KM is a discipline which promotes a collaborative approach to creation, capture, organisation, access and use of an enterprise's information assets.

⇒ KM is the process of gathering, creating, storing, distributing, sharing, using, and evaluating knowledge. It is about efficiently leveraging knowledge.

One important element is to bring knowledge into systems: knowledge based systems and expert systems.

### 3. KM Frameworks

For Nonaka tacit and explicit knowledge are not separate but mutually complementary entities. They interact with each other in the creative activities of human beings. Nonaka calls the interaction of these two forms of knowledge the knowledge conversion process.

This conversion process consists of four stages: socialization, externalization, combination and internalisation.

- The first step, **socialization**, transfers tacit knowledge between individuals through observation, imitation and practice.

Tacit → Tacit

- Personal communication and shared experiences: colleagues, clients, suppliers
- Field building, on-the-job-training  
Ex: bears catching salmon
- Between individuals

- In the next step, **externalization** is triggered by dialogue or collective reflection and relies on analogy or metaphor to translate tacit knowledge into documents and procedures.

Tacit → Explicit

- To knowledge that can be communicated: words, analogies, metaphors, models
- Dialogue
- Ex: manuals, procedures
- Between individuals in group

- **Combination** consequently reconfigures bodies of explicit knowledge through sorting, adding, combining and categorising processes and spreads it throughout an organisation.

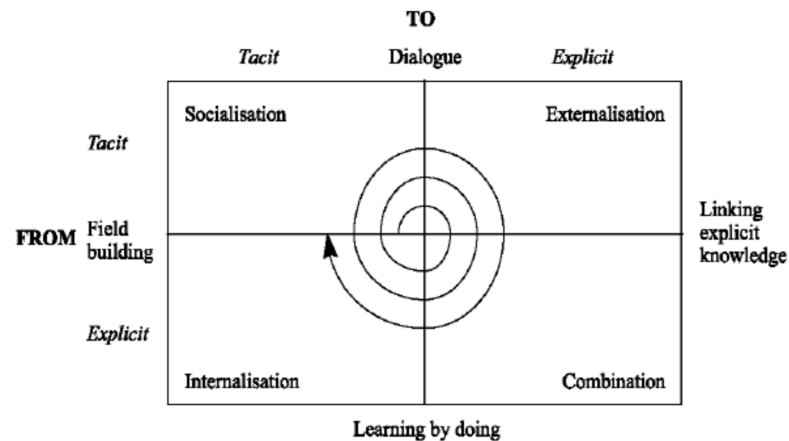
Explicit → Explicit

- Combining pieces of explicit knowledge to new whole
- Linking explicit knowledge
- Ex: literature overview
- Among groups

- Lastly, **internalisation** translates explicit knowledge into individual tacit knowledge. Eventually, through a phenomenon that Nonaka calls the "knowledge spiral", knowledge creation and sharing become part of the culture of an organisation.

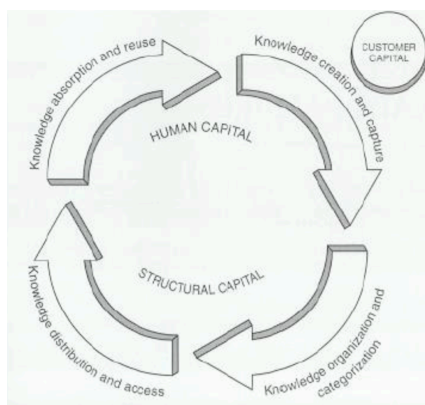
Explicit → Tacit

- Making explicit knowledge your own
- Learning by doing, experimental
- Ex: Excel learning, simulation games, Weka session
- Organization and group knowledge to individual



Nonaka (1991) emphasises that the sharing of tacit knowledge takes place through joint activities and requires physical proximity. He also states that in order for others to understand it tacit knowledge must first be externalised. Some authors feel that this is not a problem and are of the opinion that tacit knowledge is merely difficult to articulate, while others feel that although difficult to articulate tacit knowledge can be 'captured'. Others will claim that such knowledge cannot be captured and codified without becoming invalid.

⇒ Conversion of implicit to explicit knowledge and back. Organizational learning: sustaining the spiral



- Human capital: people aspect. How to get people to have more knowledge in their heads?
- Structural capital: technological aspect. How do I get the knowledge out of the people's head into an organizational asset?
- Customer capital: strength with customer. Human (relationships) or structural (products used).

- Human capital: how to get people to have more knowledge in their heads?
  - Create, capture: create a sharing culture. Ex: Buckman Laboratories
  - Absorb, reuse
    - Get knowledge in head where it can be enhanced
    - What is the difference between training and learning?
    - Recognizing knowledge brokers. Ex: The Rudy Problem
    - Foster knowledge sharing. Ex: T-shaped managers
- Structural capital: how so I get the knowledge out of the people's head into an organizational asset?
  - Knowledge organization and categorization

- Knowledge distribution and access
- Organizational capabilities needed by the market place
- Competitive intelligence
- Link with human capital
- Customer capital: how can I increase the customer mindshare?
  - Intellectual capital in minds of customers
    - Depth of knowledge of company
    - Breadth of knowledge of company
    - Loyalty
  - Think of company with great customer capital. How is this created?
  - Assist customers in learning about the company
    - Determine customer need
    - Develop sense of purpose or relationship
    - Assisting customers will increase customer capital

#### 4. KM Tools and Techniques

Knowledge management is not an ICT discipline.

- Non-ICT tools and techniques
- ICT tool and techniques

##### 4.1 Non-ICT platforms for Knowledge Management

- Culture
  - Sharing and using knowledge
  - Incentives
    - Title
    - Evaluation
  - Senior Management Involvement
- Promote knowledge sharing and using: Make sure that people are willing to share knowledge. We could for instance think of a reward for sharing. Ex: you can only download a file if you upload a document. Promote knowledge sharing and knowledge using. Ex: in Telco you want to reward people that frequently go to the FAQ questions.
- Storytelling
- Communities of practice

##### 1) Culture

Foster knowledge sharing can be facilitated through technology.

From training to learning, what is the difference?

- Pull: Employees needing help from others
- Push: overloading with information (training)

Often it should not start but end in IS department.

+ incentives



2) Storytelling. KM at WorldBank, promote the need for KM.

*"In June of last year, a health worker in a tiny town in Zambia went to the Web site of the Centres for Disease control and got the answer to a question about the treatment of malaria. Remember that this in Zambia, one of the poorest countries in the world, and it took place in a tiny place 600 km from the capital city. But the most striking thing about this picture, at least for us, is that the World Bank isn't in it. Despite our know-how on all kinds of poverty-related issues, that knowledge isn't available to the millions of people who could use it. Imagine if it were. Think what an organization we could become."*

Knowledge cannot always be reduced to analytical propositions.

An effective Tool for uncovering Tacit knowledge

- People love to read and hear stories
- Simple and effective way of conveying complex ideas
  - "An excellent way of converting tacit knowledge to explicit knowledge and an effective method for quickly assimilating new learning."
  - There is considerable evidence that story will soon become the main tool in knowledge management programmes and decision making.

3) Community of practice (CoP)

A CoP is a group of people that share their knowledge and experience on one topic. Team members are considered specialists. A CoP arises naturally.

Top into organizational knowledge

- 60-70s: centres of expertise
  - Company experts in centre
  - Experts leave
- CoPs: community is the expert  
Ex: Email, Q&A forum

CoPs are informal

- Resist being managed, cannot be designed
- Should be nurtured. How?

Nurturing CoP

- Identifying potential CoPs.
  - Cop consultants: understanding CoP, interviewing potential members on problem description and linkage with company goals.
  - Members need to personally connect!
- Providing a CoP infrastructure.
  - ICT systems: Face to face, Email, Forums, Video, conferencing, ...
  - Reward participation: evaluation, recognition as expert
- Evaluation a CoP
  - Difficult: Effects not immediate

- Listening to anecdotes how a comment resulted in a discussion, resulted in solution for major problem, new product.

#### 4.2 ICT Platforms for KM

- Codification
  - Explicit, tangible knowledge
  - Common storage
  
  - Data bases management systems
  - Search engines
  - Taxonomy and document classification
  - Wiki
  - Portals
- Personalization
  - Implicit, intangible knowledge
  - Locate each other and communicate
  
  - Expertise localization tools
  - Yellow page
  - Email, chat, forums
  - video conferencing
  - Artificial intelligence

Exam: Explain and comment following statement “Explicit knowledge does not exist, these two words are contradictive”

#### 5. Role of ICT for KM

Basic approaches

##### 1) Codification

- Explicit, tangible knowledge
- Common storage

##### 2) Personalization

- Implicit, intangible knowledge
- Locate each other & communicate

⇒ Do all companies have the same approach/use of IT for knowledge management?  
What could be important to choose either codification/personalization?

Categorized

- Product vs service-based
- Low vs. high volatility

	Low volatility	High volatility
Product-based	<ul style="list-style-type: none"> <li>•British Petroleum</li> <li>•Buckman Laboratories</li> <li>•Shell</li> </ul>	<ul style="list-style-type: none"> <li>•Hewlett Packard</li> <li>•Microsoft</li> <li>•Siemens Infineon Techn.</li> <li>•Xerox</li> </ul>
Service-based	<ul style="list-style-type: none"> <li>•Ernst and Young</li> <li>•KPMG</li> <li>•Siemens Business Services</li> </ul>	<ul style="list-style-type: none"> <li>•McKinsey</li> <li>•Skandia</li> </ul>

- **Product based, low volatility: BP, Shell, ...**

Standard products, well established processes; competition (not on basis of product alone, services that accompany product)

Knowledge for base of competition

- Intangible, fuzzy
- IT to support
  - Face to face communication
  - Expert directories
  - COP

- **Product-based, high volatility: HP, Microsoft, Siemens Infineon, Xerox**

- Highly technological products
- Competition
  - Rate of innovation
  - Speed of new product development

Knowledge for base of competition

- Product development teams. Personalisation: intangible knowledge from experts
- Sales teams: codification: explicit knowledge to sales teams
- IT to support
  - Face-to-face communication
  - Expert directories
  - Knowledge repositories

- **Service based, low volatility: Ernst & Young, KPMG, Siemens Business Technique**

- Relatively stable services
- Competition
  - Cost-effectiveness of service proposition
  - Cumulated knowledge and ability to use it

Knowledge for base of competition

- Codification: knowledge
  - Explicit
  - Stored electronically
  - Common technological platforms

- **Service based, high volatility: McKinsey, Skandia**

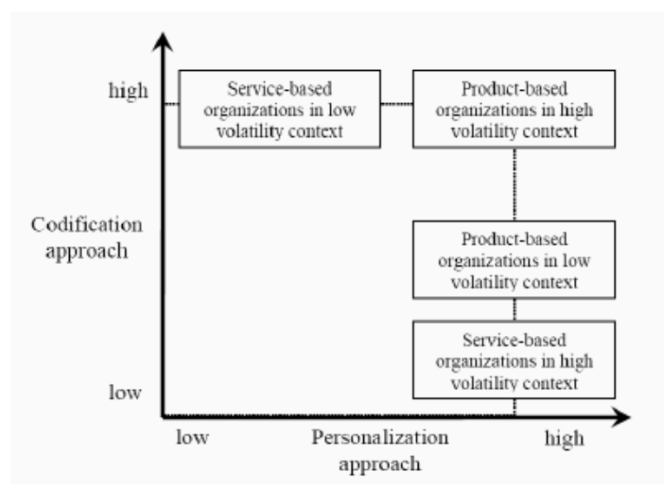
Highly dynamic, competition (tailored strategic services)

Knowledge for base of competition

- Personalization: knowledge
  - Focused on people, not IT
  - Personal communication

- Phone and video conferencing
- People transfer

	Low volatility	High volatility
Product-based	<ul style="list-style-type: none"> <li>• Strategic COP</li> <li>• Expert directories and collaborative tools</li> <li>• Reward participation</li> </ul>	<ul style="list-style-type: none"> <li>• Expert directories and collaborative tools</li> <li>• Knowledge repositories</li> <li>• Review of content</li> <li>• Reward knowledge sharing</li> </ul>
Service-based	<ul style="list-style-type: none"> <li>• Knowledge repositories</li> <li>• Effective search</li> <li>• Reward knowledge sharing</li> <li>• Reward knowledge reuse</li> </ul>	<ul style="list-style-type: none"> <li>• Culture of mutual support</li> <li>• Communication support for one-to-one interaction</li> </ul>



*Distinct patterns in use of ICT for KM*

Exam: which ICT technologies for knowledge management are suitable for Dell, University of Cambridge? Clearly state what type of organization it is, and motivate why your provided ICT technologies are suitable for that kind of company.

To conclude, knowledge is a creative process, not a physical asset. Knowledge is an ecosystem: needs nurturing, needs harvesting. It is valuable and competitive.

Knowledge management ...

- Is not a buzzword
- Is crucial in current environment
- Is not an ICT disciplinary?
- Is a cultural thing
- Takes time

## CHAPTER 17: KNOWLEDGE BASED SYSTEMS

### 1. Interfaces

Not all the knowledge comes from data. It may come from analytics, head of people, etc.

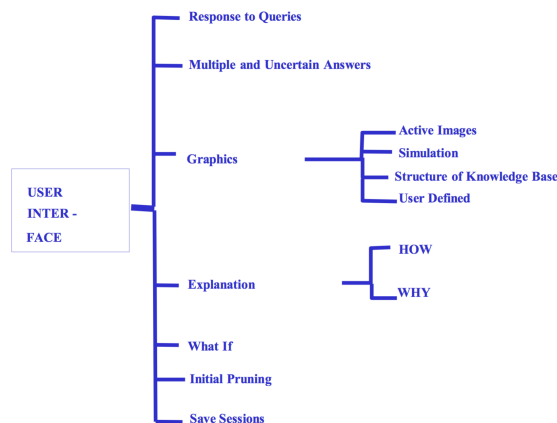
As we have seen in previous chapters, some applications are self-contained or part of another application. We then have to make sure we can use these systems: user interface. We want to make sure that knowledge is correct, we need to ensure it is complete and consistent.

⇒ How do we make sure it can operate next to other systems?

- The **user interface** allows end users to consult and work with a knowledge-based system in an efficient and correct way.
- The **system interface** must take care of a correct and effective communication with other software and hardware elements in the environment
- The **developer interface** offers a number of techniques that assist in the development process of the system.

#### 1.1 User Interface

The user interface contains items that make the system look smart. It is the fact it can deal with uncertainty.



#### Explanation

- Why question: a system should be able to explain why it asked a certain question. The user has the tendency to ask why the system wants to know this.
- How question? How did the system get the results? The reason why. How did it obtain results?  
The system produces a certain result. Then we can ask “how did you get there”. The system answers by showing the rules. The system gives some explanations about the current reasoning. The more sophisticated the system is, the more you want to know why the system did this

**Initial Pruning:** Prune the rest and immediately move to the core part.

**Save session:** start where you left things. Interfaces have a number of features; the trust of end user seems higher as they can ask questions and receive feedback. Trust is obtained using smart dialogs.

### Use of explanation facilities

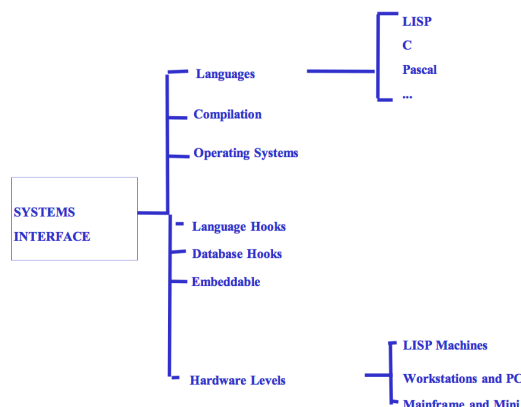
⇒ Clarify things and make it easier for the users to deal with the system.

- 1) Assisting in **debugging** the system
- 2) **Informing** the user of the **status** of the system
- 3) Increasing the user's **confidence** in the system
- 4) **Clarification** of terms and concepts used by the system
- 5) Increasing the users **personal level of expertise**

## 1.2 System Interface

These are the mean a knowledge based system has to interface to normal systems. It can't work on its own. It has to connect to other systems. Ex: how to calculate the salary of employees. The system may need to connect to the employee database.

The system must be able to call other systems.

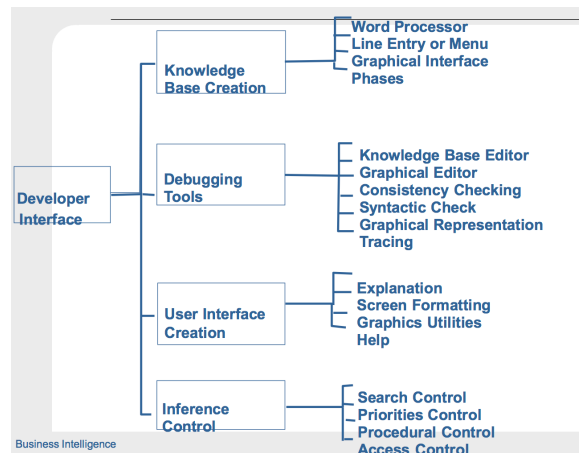


The system must be able to call other systems. In a database you have a number of features. Copy all data to the rule based system and reason with it. Is this possible? May be a bit naïve.

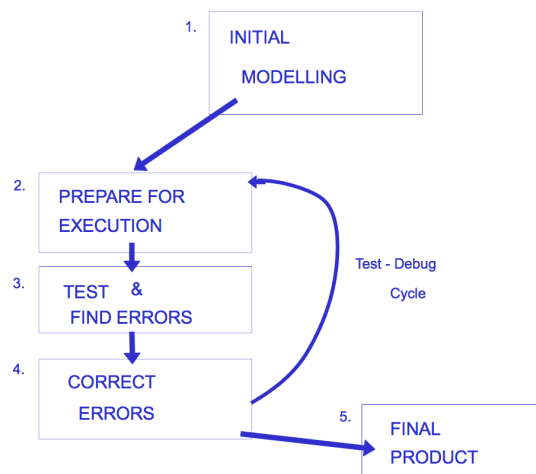
- **Internal data management**  
Ex: Prolog (but: volume, support); prolog with DBMS enhancements (indexing, buffering, ...)
- **Static (loose) coupling** with existing DBMS. Snapshot copy at start (volume, redundancy).
- **Dynamic:** reason with a set of data at the same time.
  - Interface to (relational) DBMS
  - Optimizing interface to (relational) DBMS
  - Integrated interface an intelligent database machine

## 1.3 Developer Interface

Build a system based on a set of rules. Business has so many rules and laws. It is hard to make sure a set of rules is complete, consistent and correct. It tries to make it easy to build a system.



Debugging and quality tools are more interesting for us. Test-debut cycle is a dangerous approach. We try to build a system on the fly and we improve it when required.



⇒ It remains too hard to add rules to a system to make it work.

## 2.Verification and Validation

Make sure the system is complete, consistent and correct. We want the system to be correct all the time, not from time to time. There are different levels of correctness and errors

- Syntactical mistakes
- Logical mistakes
- Invalid knowledge

Verification and validation are two distinct terms we should not mix up.

- Verification: Do we build the system in the right way? Did we use the correct steps? This step can be done by anyone.

- **Validation** is about requirements. Do we build the right system? Does the system correspond to the requirements? Do we build the right system? Validation can only be done by someone familiar with the domain.

There are different types of faults

- **Factual faults:** an assertion does not correctly represent the real fact.  
Ex: A car has five wheels.
- **Inferential faults:** A rule does not correctly represent the domain knowledge. The result is that incorrect conclusions are drawn by the system.
- **Control faults:** the rules are correct but have undesirable control behaviour.

Author	Verification	Validation
Green & Keynes	A “paper” activity showing that the code fully and exclusively implements requirement specifications	A “live activity” showing that the code satisfies user requirements
Stachowitz & Coombs		Shows correctness with respect to specifications
Culbert, Riley & Saveley	Complete with respect to requirements of previous development phase	To ensure final compliance with software requirements.
Martin-Mattei	Complete with respect to requirements of previous development phase	To ensure finally compliance with software requirements; subsumes methods for checking, consistency, exactness, and completeness under formal validation.
Geissmann & Schultz	Shows that the system is developed correctly and does not contain technical errors	Ensures that the system satisfies user needs and is usable for intended purpose
Naser	Review of design with respect to requirements for determining whether the right problem is solved	Covers testing and evaluation, shows compliance with functional requirements, and ensures that detected errors are corrected
Landauer	Comparison of rule base with specifications of desired behaviour	Comparison of specifications against external notion of correctness

*Overview of definitions of V&V (Hoppe & Meseguer, 1993)*

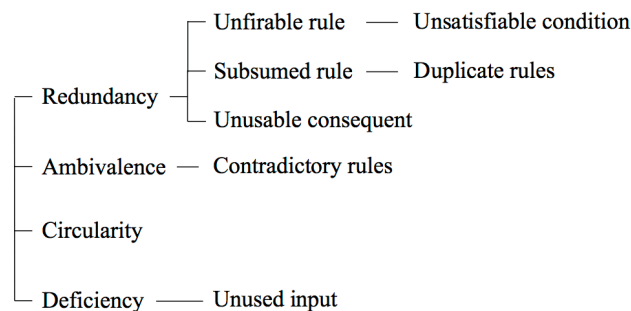
### Definitions

- **Verification:** checks a KBS against the specifications generated by its totally formalizable requirements. It is performed by checking and not by proving KBS properties.
- **Validation:** it assures that the final KBS complies with user needs and requirements. Validity can only be achieved partially by verification, since some informal requirements can only be partially formalized. Therefore, specific methods are required to check informal requirements and should be present during the entire KBS life cycle.
- **Testing:** performed by executing the KBS on test cases and analysing the results. This can be considered a comparison of KBS behaviour against a specification of intended behaviour expressed as test cases.
- **Evaluation:** assesses or measures a KBS’s quantitative and qualitative characteristics and compares them with expected or desired values.

### **Rule anomalies**



If you have a number of rules, some things may go wrong: inconsistent rules, redundant or deficient rules, etc. How to make sure we do not have missing rules?



*Preece's anomaly classification*

### Consistency of the Rule Base

- **Redundancies (Maintenance, Uncertainty)**
  - Redundancy
  - Subsumption
  - Unnecessary conditions
- **Inconsistencies**
  - Conflict (contradictory conclusions)
  - Cyclical Rules
  - Dead end rules
  - Illegal attributes, values, ...

### Completeness of the Rule Base

- **Missing knowledge**
  - Missing knowledge areas
- **Incomplete knowledge**
  - Missing rules (missing attributes/combinations)
  - Dead end conclusions
  - Dead end premises

1) Redundancy: something is double in the knowledge base. It is not necessary.

A rule  $R$  is redundant in  $\mathcal{H} \Leftrightarrow \forall e \in \mathcal{E}, \forall g \in G:$

if  $\mathcal{H} \cup e \rightarrow g$  then  $\mathcal{H} \setminus \{R\} \cup e \rightarrow g$  ■

Three types of redundancy may occur in a rule base, i.e. unfirable rules, subsumted rules and unusable consequent.

- An **unfirable** rule is a rule which has no firable instance. This means that the antecedent of the rule can never be satisfied.  
Ex: If RegionWine = Bordeaux AND Regionwine = Burgundy THEN Origin = France.
- **Subsumption** occurs when the second rule is more general than the first rule. The first rule is then redundant; a special case of subsumed rules

are duplicate rules. These are rules of which the antecedent and the consequent are identical.

Ex: IF Capital = Brussels AND city = Antwerp THEN Country = Belgium. IF Capital = Brussels THEN Country = Belgium

**Subsumed Rules:**

Rule 1. IF Firm's net working capital > 0  
AND trend in net working capital is  
negative THEN solvency rating is low.

Rule 2. IF Firm's net working capital > 0  
AND trend in net working capital is  
negative AND firm's current ratio < 2  
THEN solvency rating is low.

- **Unnecessary conditions**

**Unnecessary IF conditions:**

Rule 1. IF Firm's net working capital > 0  
AND trend in net working capital is  
negative THEN solvency rating is low.

Rule 2. IF Firm's net working capital < 0  
AND trend in net working capital is  
negative THEN solvency rating is low.

**New Rule** IF trend in net working capital is  
negative THEN solvency rating is low.

- **Unusable consequent:** this type of redundancy occurs when a literal appears only in the consequent of a rule, indicating that this literal should be a final conclusion.

Ex: IF Capital = Brussels THEN country = Belgium

- 2) A rule base is **ambivalent** if it is possible to infer semantic constraints. Ambivalence is also commonly denoted as inconsistency. A special case of ambivalence are contradictory rules. In this case, opposite conclusions can be inferred from the KBS.

Ex: IF Capital = Brussels THEN Country = Belgium ; IF Capital = Brussels THEN Country = Germany

**Contradiction:**

Rule 1. IF Solvency rating is average AND  
profitability rating is average THEN apply  
for a secured loan.

Rule2. IF Solvency rating is average AND  
profitability rating is average THEN apply  
for an unsecured loan.

- 3) **Circularity** occurs in a KBS when it contains a set of rules, which can create a loop when the rules are fired. For instance, the following rule base might depending on the inference mechanism, never stop with making the inference, if one rule is fired. Each rule will trigger the other rule to fire, such that an endless loop will be created.

Ex: IF Capital = Brussels THEN Country = Belgium ; If Country = Belgium THEN Capital = Brussels.

**Circular Rules:**

Rule 1. IF management competence is good  
AND financial credit rating is good THEN  
overall credit rating is good.

Rule 2. IF overall credit rating is good AND  
trend in profitability is very good THEN  
management competence is good

- 4) **Deficiency** occurs in a rule base, if there exists a combination of literals for which a conclusion that should be inferred is not inferred. The problem of deficiency is commonly known as the problem of incompleteness. For example, if the set of possible inputs for capital is (Brussels, Bonn) and the rule base consists of the following rule: IF Capital = Brussels THEN Country = Belgium.

**Missing Rules:** suppose Management Competence is High, Low or Medium

Rule 1. IF financial analysis rating is high  
THEN management competence is Low.

Rule 2. IF financial analysis rating is low  
THEN management competence is High.

A special case of deficiency is an **unused literal**. An unused literal is a literal which has been defined in the specifications, but this literal does not occur in any rule in the knowledge base. Ex: if the literal Pregnant occurs in the specifications and pregnant is not used in the rules, then this literal is an unused input.

To detect the anomalies, many tools have been developed. Currently, tools can adopt two strategies to analyse a knowledge base.

- Either they use **meta-knowledge** to check the system (domain dependent tools)
- Or they transform the knowledge base in an intermediate representation, such as a table or a graph (domain independent tools)

An example of a tool that uses meta-knowledge is the Expert System Validation Associate system. This system was developed at Lockheed Corporation. Eva is a set of tools built around a theorem prover and a database. These tools include anomaly checks and validation tools. Eva works as a front end to several shells and transforms the syntax of those shells into Eva format. The meta-language of Eva allows the knowledge engineer to specify semantic constraints (e.g., impermissible events).

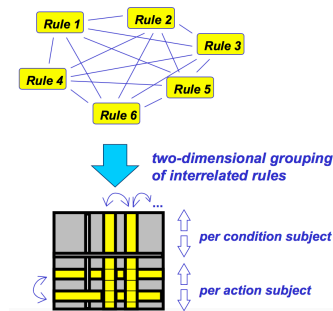
A second category of tools are the domain independent tools. Early approaches in V&V made use of some form of tables. Examples of these tools include the Rule Checking Program, Expert System Checker, and Puuronen's approach. A disadvantage of these tools is that they merely check anomalies between pairs of rules, no checks over chains

of rules are carried out. Second generation methods make it possible to detect anomalies across numerous rules.

In RCP, checks are built in to detect anomalies between pairs of rules which conclude the same value for a parameter in the same context. The following procedure is executed to check the rules.

- Find all parameters used in the conditions of these rules.
- Make a table, displaying all possible combinations of condition parameter values and the corresponding values that will be concluded for the actions parameters.
- Check the tables for conflict, redundancy, Subsumption and missing rules. Then display the table with a summary of any potential errors that were found. The rule checker assumes that there should be a rule for each possible combination of values of condition parameters; it hypothesizes missing rules on this assumption.

Nowadays, we start from rules and then build decision tables. If you build a correct decision table, no validation step is needed anymore.



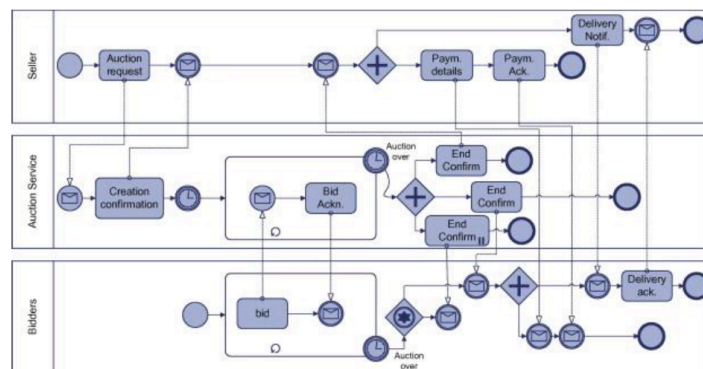
## CHAPTER 18: DECISIONS, RULES AND PROCESSES

### 1. Introduction

⇒ Interesting to compare rules, models and processes using some criteria's.

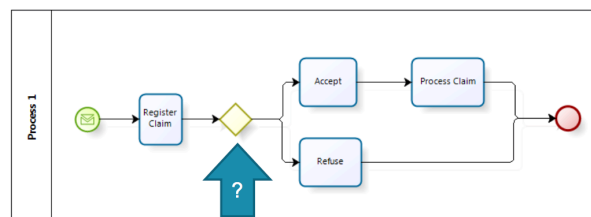
Declarative BPM starts from rules and constraints. In process mining we have imperative mining, rule mining and mixed forms of mining.

⇒ What should be in the process model? how much is in a process? Do we include all the rules in this? Exceptions? Timers? Happy path? Decisions? Decision logic? Roles? Messages? Notifications? Triggers? Conditions? ... How to combine rules and process? How to integrate them?



Decisions are important for business, not only processes. Why would we only model the processes? Where is the decisions? How is the decision log modelled?

Model the decision activity: Decide acceptance



How much details do I put in the model? Activities, pools, lanes, decisions, rules, ... We cannot have a detailed and general map at the same time.

Look at the process, do we have all the paths? Is it complete? Is it consistent? Almost impossible to check this in a process model.

It is important to be aware of the real challenges. Use the correct representation and use a set of techniques to do verification.

### 2. Decision Model & Notation

## 2.1 Primary Use Cases

- Modeling human decision-making
  - Descriptively model the decisions within an organization
  - Natural language used rather than formal expressions
- Modeling requirements for automated decision-making
  - Descriptively model the decisions within an organization
  - Notation for Modeling Decisions, Decision Table Types
  - Formal expressions used but may be incompletely specified
  - Some decisions may be delegated to humans
- Implementing automated decision-making
  - Decision logic must be completely specified
  - Friendly Enough Expression Language (FEEL)

## 2.2 DMN Levels

### Decision requirements level

The decision requirements level RDM of a decision model DM is a single decision requirement graph depicted as a set of decisions requirement diagrams.

Decision requirement diagrams (DRD) denote the information requirements of each decision, by connecting them with their sub-decisions and inputs. This is represented by a directed acyclic graph. The DMN specification allows a DRD to be an incomplete or partial representation of the decision requirements in a decision model.

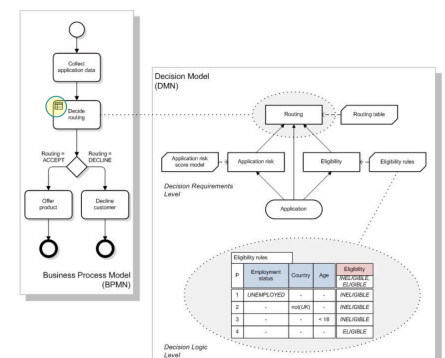
### Decision logic level

A decision is the description of the decision logic used to determine an output from a number of inputs.

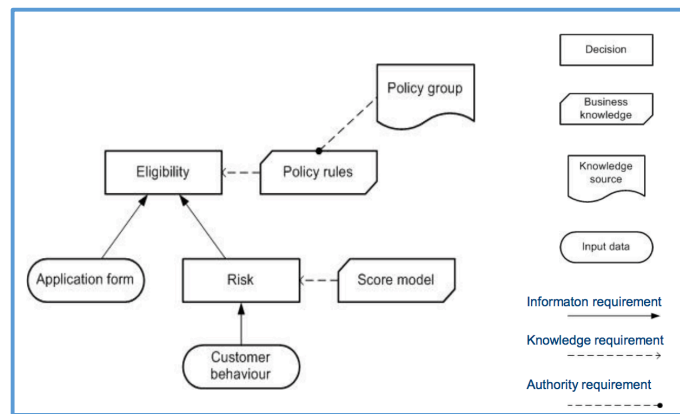
## 2.3 Decision Requirement Graph

- **Property 1.** Given a decision model every decision  $D$  in that model has a unique decision requirement graph DRGD with  $D$  as its single top-level decision.
- **Property 2.** The topological order of a DRD induces a partial order  $\leq$  on the decisions contained in the DRD.

For two decisions  $D1$  and  $D2$  we say  $D2 \leq D1$  if and only if there is a directed path from  $D2$  to  $D1$ , i.e.  $D2$  is a sub-decision of  $D1$ .



Since decisions are declarative, this partial order does not dictate an execution order, but rather a requirement order.



## 2.4 Decision Table

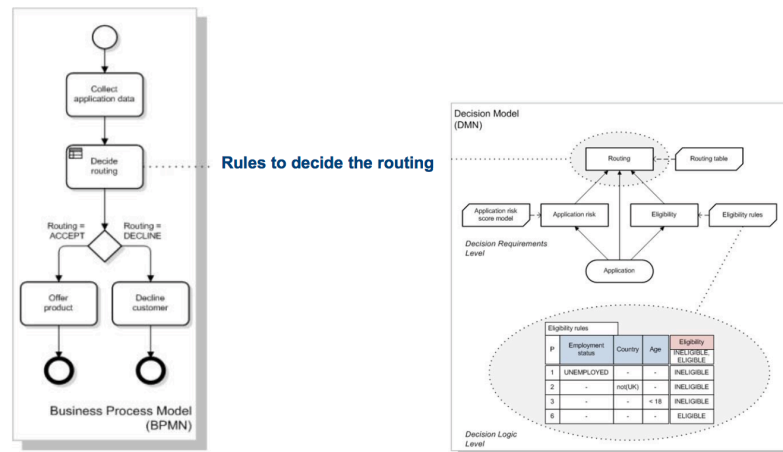
Decision tables are a precise yet compact way to model complex rule sets and their corresponding actions.

Decision tables, like flowcharts, if-then-else, and switch-case statements, associate conditions with actions to perform, but in many cases do so in a more elegant way. Each decision corresponds to a variable, relation or predicate whose possible values are listed among the condition alternatives. Each action is a procedure or operation to perform, and the entries specify whether (or in what order) the action is to be performed for the set of condition alternatives the entry corresponds to. Many decision tables include in their condition alternatives the don't care symbol, a hyphen. Using don't cares can simplify decision tables, especially when a given condition has little influence on the actions to be performed. In some cases, entire conditions thought to be important initially are found to be irrelevant when none of the conditions influence which actions are performed.

Routing				
U	Eligibility	Application Risk	Age	Routing
1	INELIGIBLE	-	-	DECLINE
2	ELIGIBLE	HIGH	-	DECLINE
3		MODERATE	< 25	DECLINE
4			>= 25	ACCEPT
5		LOW	-	ACCEPT

## 3. Decisions and processes: the role of decision models

- 1) A Decision Model Corresponding to a Single Decision Activity in a Process Model  
When there is no interleaving of data throughout the mode, decisions can be used in this form of single branches that use the output of an activity that has implemented and evaluated the criteria for making the decision.

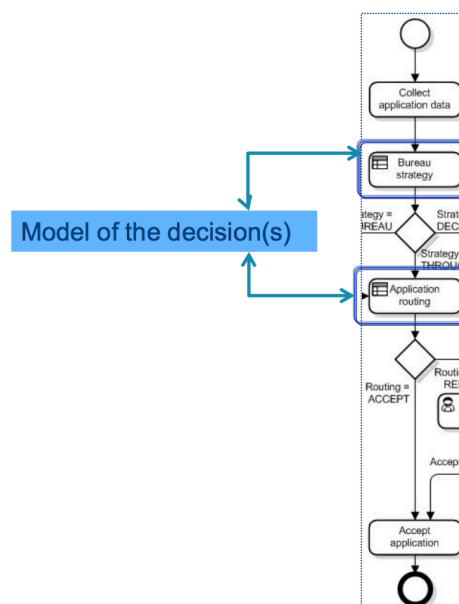


## 2) A decision Model Spanning over Multiple Decision Activities in an Existing Process Model

Multiple activities in a process model may refer to different decisions that are all part of the same decision model. Are the data needed as input for a decision available before the decision is invoked? Activities producing the input required by decisions should occur before invoking those decisions. Otherwise, situations appear in which decisions cannot be made consistently due to incomplete or incorrect input.

When decisions are dependent on intermediary results that are spawned earlier in the process, the decision model also puts constraints on the sequence ordering of the workflow.

Decision models are not lower level details of one process. They can span over multiple activities, and even multiple processes. Separation of concerns.



## 3) A Decision Model That Can Be Translated to a Straightforward Process for Execution. Sometimes the business process is really about a big decision.

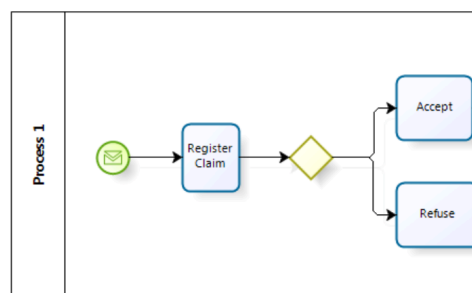


Some approaches model this in business process models, hence forgoing the purpose of decision models that were designed specifically for this task.

Other approaches rather seek to find the balance between data-driven models and business processes. However, the process part still remains a subordinate to the decision model.

In this paper we assume that when the process is really about a big decision, the process model can be considered as the chosen execution flow to make the decision.

Sometimes, the entire process is about a decision. Model the decision first, and then think about how to execute it. The same decision can be processed in many ways. The process of making a decision depends on the desired criteria (throughput, efficiency, customer comfort, ...).



- 4) Executing Decision Model Beyond One Fixed Decision: Flexibility Once a decision model is built, it could be used for multiple purposes, not just the obvious decision that is present in the context of a current business question.

The decision model could be designed for the current process, but also for other or future processes.

#### 4. Execution scenarios

##### 4.1 Execution Scenario 1

###### ⇒ Standard Forward Decision Execution

The decision logic is captured in a straightforward decision model and the decision is to find the correct outcome for a specific set of input values. These are typical current DMN applications, e.g., determine the discount for specific clients, based on discount policy, client data, history, etc. or determine eligibility for insurance given the company policy.

The reasoning mechanism does not need high flexibility, but it is important that company rules and policies can easily be adapted and brought to implementation. The routing decision with the associated decision table shown earlier is completely invokable when all its input is available.

Routing				
U	Eligibility	Application Risk	Age	Routing

## 4.2 Execution Scenario 2

Often the decisions inputs are not simply available up front, but can be obtained at a certain cost (database lookup, user question). The decision is still to find the correct outcome for a specific set of input values. But the order of looking up input data and answers to user questions can have cost implications.

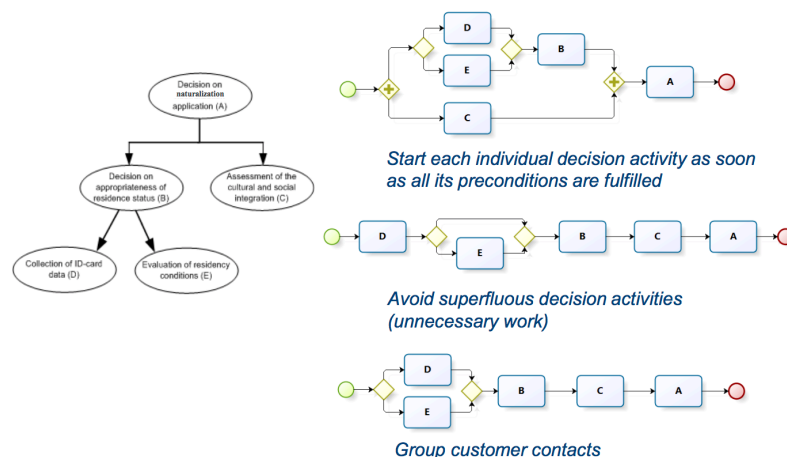
If a decision can be made with only a partial set of inputs, it is more cost effective. The decision model is the same, but the execution of the specific decision might be optimized.

The question here is: what is the optimal process of executing the specific decision, given the cost of obtaining data and the frequency of cases?

As various process models can result from a decision model, the choice between different process models becomes an important issue. Possible optimization criteria are :

- Customer perspective: minimal points of contact
- Business process behavioural perspective: starting with a labor-intensive activity is not optimal
- Organizational perspective: minimizing the number of handovers
- Informational perspective: all necessary information could be easily acquired at one point in time.
- External environmental perspective: assessing external information should be limited.

With various decision process models to choose from, the decision process model that fits best with the business requirements can be chosen.



These criteria allow for two types of optimization, one **static**, the other **dynamic**. The approaches depend on the assumption made about input availability.

- **Static Optimization** of Forward Decision Execution. In the static approach the assumption is that all input is available (at a cost). The decisions execution can then be optimized by ordering the different sub decisions using the before mentioned criteria, or the cost, or frequency of these decisions.
- **Dynamic Optimization** of Forward Decision Execution. Apart from the previous static approach, a dynamic approach can be taken if the assumption that all input is available up front is invalid. Using the above criteria and the

requirement order of the decision model, an optimal process for the execution of the decision can be generated dynamically on a case-by-case basis.

#### 4.3 Execution Scenario 3

The specific decision to be answered is not always fixed.

When dealing with a credit application, for example, sometimes the decision is not: Does this specific customer (with all input data available) get a loan?, but: What can we already derive from the available data?

Another question could be: Which changes should be recommended to the customer to obtain a higher loan?

DMN decisions do not specify whether the required input data is needed up front to be able to determine the outcome and assume a given decision path.

Routing				
U	Eligibility	Application Risk	Age	Routing
1	INELIGIBLE	-	-	DECLINE
2	ELIGIBLE	HIGH	-	DECLINE
3		MODERATE	< 25	DECLINE
4			>= 25	ACCEPT
5		LOW	-	ACCEPT

To conclude ...

- Complement business (process) models with decision modelling
- Create flexible decision execution
- Optimizing the execution of decision models
- Many inference techniques
- Incomplete data
- Influence on process modelling
- Comprehensible and expressive declarative specifications

## CHAPTER 18: ARTIFICIAL NEURAL NETWORKS AND SUPPORT VECTOR MACHINES

We have seen some methods where we derive some rules from a data set. But there are some disadvantages with decision tree induction. In order to improve the accuracy of decision trees we will construct larger trees.

To deal with disadvantages, artificial neural networks and later, SVM's will allow us to come with an accuracy which is usually higher. But one of the disadvantages will be that the result is a model, a black box. It is a model that classify cases, but it is somewhat harder to understand the model. Hence, there will be a trade-off as NN and SVMs increase accuracy but decrease the overall Understandability.

### 1. Neural Network

#### 1.1 Characteristics

A Neural Network is an information system that identifies objects or patterns based on examples that have been used to train it. It is often used to solve problems where formulas or procedures for solving are not defined.

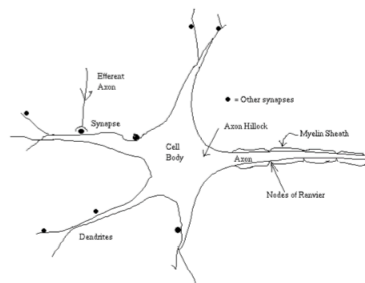
Artificial neural networks mimic the working of the human brain. It is basically some kind of model that identifies patterns based on examples and that has been trained to learn from the examples. The knowledge is not known, neither defined nor written down. We only try to learn from examples.

We have loads of data and outcomes we want to learn from.

It works best where a large database of examples is present and where even experts cannot supply rules linking inputs to outputs.

Ex:

- Handwriting recognition
- Business failure prediction
- Time-series prediction (Ex: share prices)



The name Neural Network comes from how the brain more or less works. It is supposed the human brain receives stimuli, react on these and then surpass a certain value if there are more stimuli.

- Neurons: brain cells
- Nucleus (at the centre)
- Dendrites provide inputs
- Axons send outputs

⇒ Interconnected Neurons exchange information

One of the definitions of artificial intelligence was to mimic or to understand the human brain. The first NN was modelled to see how the human brain works.

The way human brain learns is highly **parallel**.

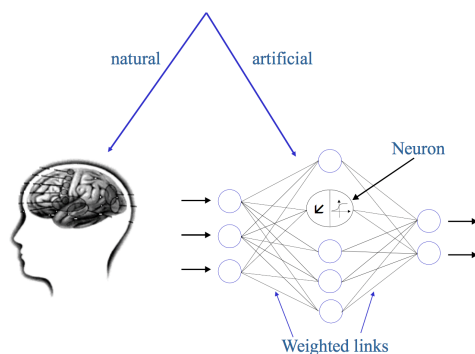
Once we have our examples we can learn from it in two ways

- **Unsupervised**: we have the data but we have no outcome
- **Supervised**: we have data with outcome from the past and we learn how to obtain the outcome. We are looking for the mechanism to go from inputs to outputs and we try to discover patterns using simple or multilayer Neural Networks.

Ex: Multilayer Perceptron

- Classification
- Regression

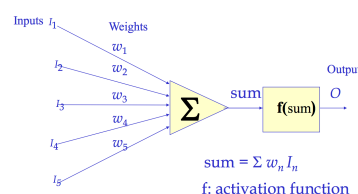
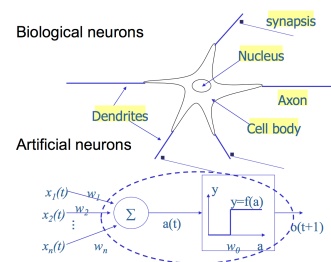
## 1.2 The Neuron model



Weights are important things. We try to discover how to weigh different inputs and how to transform inputs to the different outputs.

The transformation from the input to output will be done through an activation function which might have different forms.

We have a cell, which as a nucleus, that receives inputs through axons and dendrites. We have some inputs, we have some weights and we make the weighted sum of all the inputs. If the sum is larger than a certain value, we say “yes” otherwise we say “no”. This is the basic mechanism.



The **activation function** works on the sum of the weighted inputs. By training the neural network, we will adjust the weights, based on what we have observed. The difference between the actual outcome and the desired outcome is used to feed back into the weights of the network. This is **back propagation** learning: adjusting the weights.

⇒ The activation function can be linear, logistic, hyperbolic, any form.

logistic (sigmoid)

- ◆  $f(\text{sum}) = 1/(1+\exp(-\text{sum}))$
- ◆ between 0 and 1

hyperbolic tangent

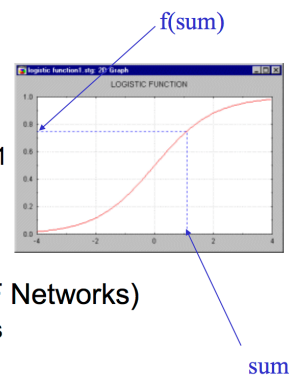
- ◆  $f(\text{sum}) = 2 * [1/(1+\exp(-\text{sum}))] - 1$
- ◆ between -1 and 1

linear

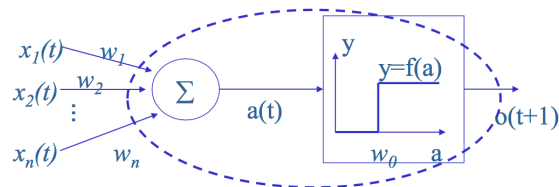
- ◆  $f(\text{sum}) = \text{sum}$

Radial Basis Function (RBF Networks)

- ◆ Gaussian activation functions



Weights are not fixed but are exactly what has to be learned or adjusted. So we need a way to adjust the weights. This is called a **learning rule**. We feedback from the outcome to the earliest stages in the model.



Learning rule: Donald Hebb (1949): The Organization of Behavior.

Variable weights ⇒ Neurons learn a function.

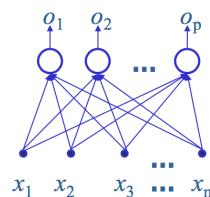
$$W_{ij} = x_i * x_j$$

$W_{ij}$ : weight between neuron i and neuron j

$x_i$  ( $x_j$ ): activation of neuron i (j)

### 1.3 Learning with perceptron

The simplest form of neural network: 2-layer neural network. We have some inputs; we take the sum of the inputs. We take the function and obtain a result. It is nothing more than a weighted sum of the inputs. This is called a **perceptron**.



$$o_i = f(a_i) = f\left(\sum_{k=0}^n w_{ik} x_k\right) \quad i = 1, \dots, p$$

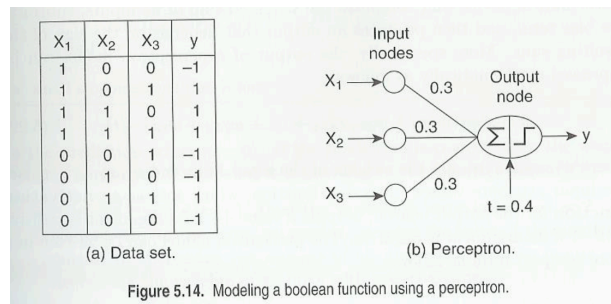


Figure 5.14. Modeling a boolean function using a perceptron.

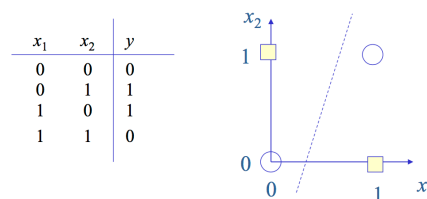
### Outcome

- 1: if at least 2 of the 3 inputs are 1
- -1 if less than 2 of the inputs are 1

We want to build a small neural network. I give a weight to each of the three inputs. And I take the weighted sum of the inputs to get my final outcome. If my weighted sum is larger than 0.3 (or 0.4), then the result is 1, otherwise the result is -1. Nothing is learned here, the model is just built.

But Neural Network were shown not to be able to solve the “XOR” problem. Basically, the result is a 1 if one input and only one input is 1. If both inputs are yes, then the result is 0. The same goes if no input is “yes”.

This means that the 2 yellow blocks are yes. The empty circles result in a null.



Minsky, Papert (1969)

$$\begin{aligned} w_1 &= ? \\ w_2 &= ? \\ \theta &= ? \end{aligned}$$

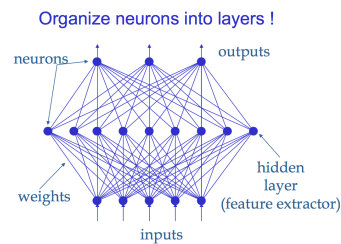
⇒ Draw a line to separate the “yes” and “no” cases. But there is no such a line. So this is not a very powerful technique since we cannot solve such a basic problem.

What if we would have an additional layer? This means we add a layer of neurons between the inputs and outputs, leading to a **multilayer neural network**, which is the same principle. We have some inputs; We have some weights. This MLP lead to the success of Data mining.

The majority of neural network applications are MLP. They allow us to do some non-linear classification and regression. Instead of drawing a straight line, we now draw a curved line to separate positive and negative examples.

$$o_i = f\left(\sum_{j=0}^n W_j f\left(\sum_{k=0}^n w_{ik} x_k\right)\right)$$

- ⇒ How do we learn from historical cases? Adjust weights. We work towards the leftmost set of neurons; We then learn by propagating the differences backwards to the front.



In this way we should be able to solve the XOR problem.

#### 1.4 Backpropagation learning

- ⇒ How does a tool learn? It learns by adjusting the weights.

Weights are the parameters that need to be estimated (cfr. Regression coefficients). Weights are randomly initialised and adapted during learning or training based on examples.

But: Neural Networks can learn the wrong things!

Provide networks with training examples one by one

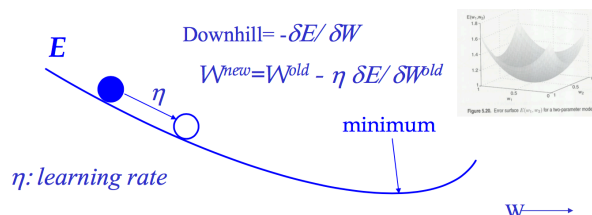
- If output is **correct**, no adaptation of weights
- If **not**, back propagate errors through the network and adapt weights according to the learning rule
- Keep providing network with training examples until stop criterion is met

Delta learning rule: minimise  $E = \frac{1}{2} \sum_n (o_n - t_n)^2$

N: number of training examples; o output of network; t target output (supervised learning!)

- ⇒ We learn by **adjusting** the weights. By propagating the difference between the real output (= what we see) and target output (= what we should have seen). The difference is used in training the weights, we sum all the errors and we then try to minimise the error term during the learning process.

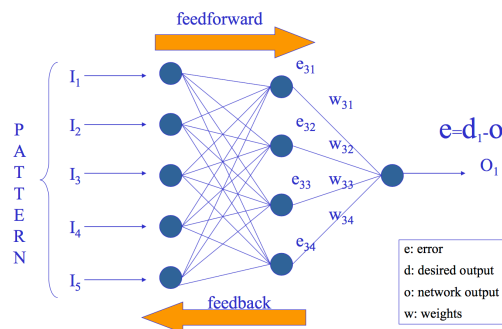
The error E is a function of the weights and the training data. If you want to minimise the error term, use the **steepest descent**. We always go in the direction with the highest decrease, this is the most promising direction.





Steepest descent is a heuristic; It is not guaranteed to provide us with the optimal result. It can bring us in a local optimum.

Visually, we have a number of inputs, nodes, multiple layers of nodes. We calculate the difference between the desired and real outcome. This error will be minimised by adjusting the weights, giving us a set of weights for the next training case.



⇒ How do we train?

- **Batch mode:** pass all training instances through the network and update weights afterwards.
- **On-line mode:** pass training instances one by one and update weights. Adjust the weights for every new example.

One epoch = 1 pass of training data through network.

Multiple epochs are needed to minimise Error E.

When do we **stop**?

- When the error is below a certain value
- When we have taken so much time over number of runs which is so high
- Error on validation set increases: we are overtraining the data.

⇒ Stop when you are confident you have learned enough.

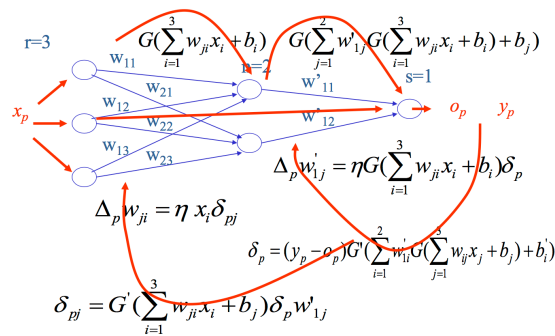
Problems:

- Gradient descent can end up in local minimum (multimodality of the Error function E)
- How to choose the learning rate  $\eta$ 
  - Small  $\eta$  : small steps in Error space, slow convergence
  - Large  $\eta$  : fast convergence but we may miss the minimum.

⇒ **Adaptive learning** rate, start with large  $\eta$  and decrease gradually. Take small steps when you come closer to the goal.

**Solutions**

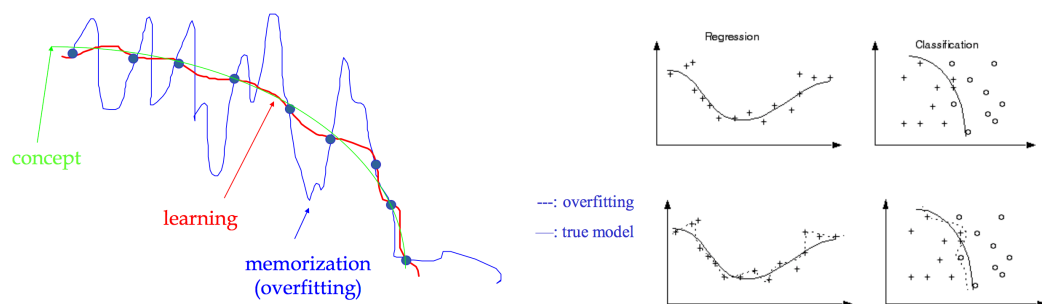
- New and advanced training algorithms. Ex: levenberg-Marquardt, Conjugate-gradient
- Support Vector Machine: no local minimum instead 1 global minimum. Convex optimisation.



### 1.5 Learning Vs. Memorisation

We train the model based on the data we have, which is a limited set of data. Training on the data we have might risk to not learn the concept or the pattern but learn the data itself, meaning we overfit or memorize the data, instead of learning the concept behind this.

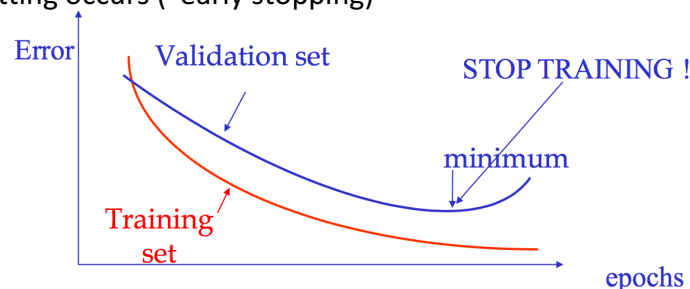
- **Successful learning:** recognize data outside the training set, i.e. data in an independent test set.
- **Memorization (overfitting)**
  - Each data set is characterized by noise (idiosyncrasies) due to incorrect entries, human errors, irrationalities, noisy sensors, ...
  - A network that is too complex (e.g. with too many neurons) may fit the noise, not just the signal leading to overfitting.

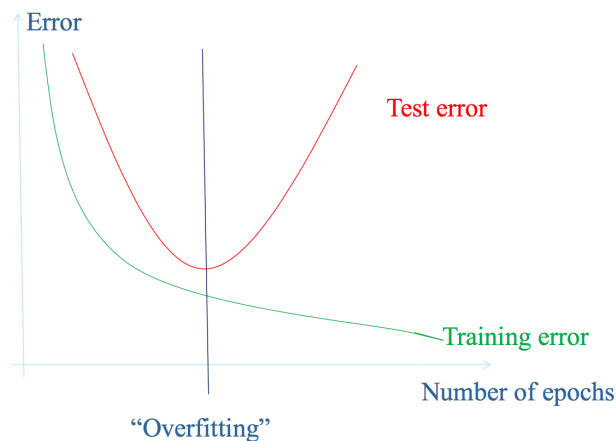


⇒ You don't want to learn the noise; You want to learn the concept behind the data.

How to avoid overfitting?

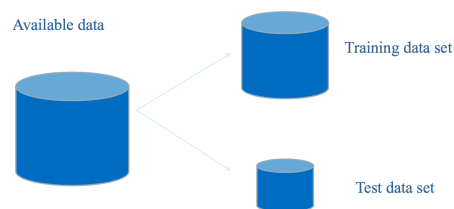
- Set aside a validation set
- Use training set to train weights and validation set to stop training when overfitting occurs ("early stopping")





We are running our cycles; We have our error terms. We stop learning when error goes up on another set of data.

So, basically we separate the available data into a training set and a test set (validation set).



Reduce model complexity

- Less complex models are less sensitive to overfitting
- Reduce the number of hidden neurons, layers, ...
- Prune individual connections from the network (OBD, OBS, ...)



Use weight regularisation terms in the Error Function  $E$

- Large weight values lead to overfitting
- Extend the error function  $E$  to penalise large weights

⇒ How many hidden neurons?

- "A rule of thumb is for the size of this hidden layer to be somewhere between the input layer size and the output layer size ..."
- "How large should the hidden layer be? One rule of thumb is that it should never be more than twice as large as the input layer..."
- "Typically, we specify as many hidden nodes as dimensions needed to capture 70-90% of the variance of the input data set..."

It is argued that the best number of hidden units depends in a complex way on:

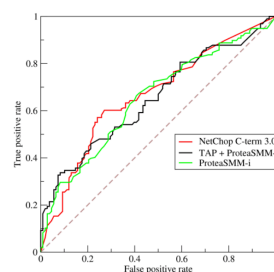
- the numbers of input and output units,
- the number of training cases,
- the amount of noise in the targets,
- the complexity of the function or classification to be learned,
- the architecture,
- the type of hidden unit activation function, the training algorithm, etc.

### Validation of classification models

- Confusion matrix. n Accuracy.
- Sensitivity.
- Specificity.
- Area under Curve (AUC): Area under the ROC curve.
- Gains chart.

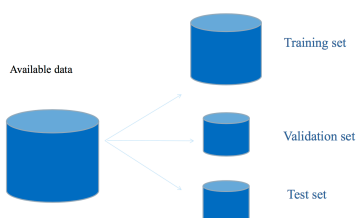
		Reality		
Model		Class 1 (positive)	Class 2 (negative)	
	Class 1 (positive)	True positive (TP)	False positive (FP) (Error I)	$P' = TP + FP$
	Class 2 (negative)	False negative (FN) (Error II)	True negative (TN)	$N' = FN + TN$
		$P = TP + FN$	$N = FP + TN$	

True positive rate (TPR) (= sensitivity) =  $TP / (TP + FN)$   
 False positive rate (FPR) =  $FP / (FP + TN)$   
 True negative rate (TNR) (= specificity) =  $TN / (FP + TN) = 1 - FPR$   
 Accuracy =  $(TP + TN) / (P + N)$   
 F1 score =  $2TP / (2TP + FN + FP)$

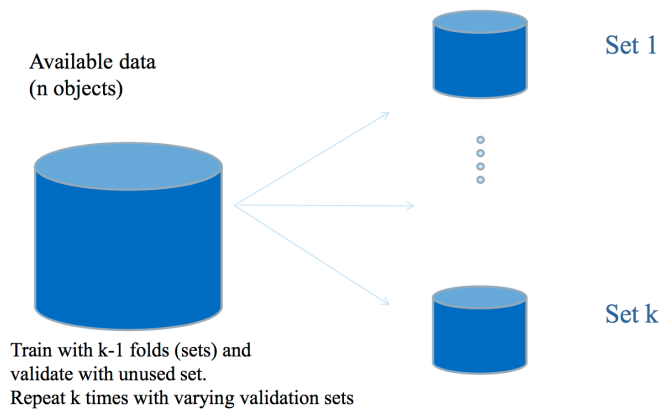


AUC (Area under curve) =  
 The area under the curve is the percentage of randomly drawn pairs for which the test correctly classifies the two samples which have been randomly chosen from each group.

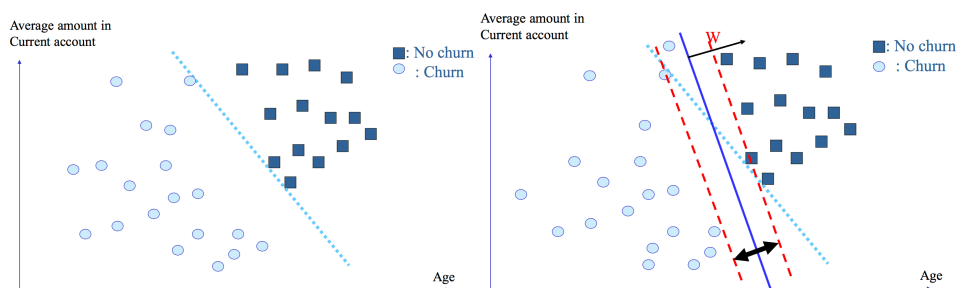
### Methodology for model selection



### Methodology to validate analytical models: k-fold cross validation

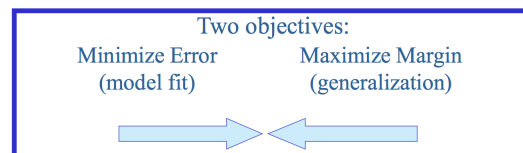


## 2. Motivation SVM



Find a classifier which:

- Minimizes the error in the training set
- Maximizes the separating margin (i.e. improves generalization)



$n$  objects :  $x_i \in \mathbb{R}^m, i=1, \dots, n$  and their respective labels  $y_i \in \{-1, 1\}$

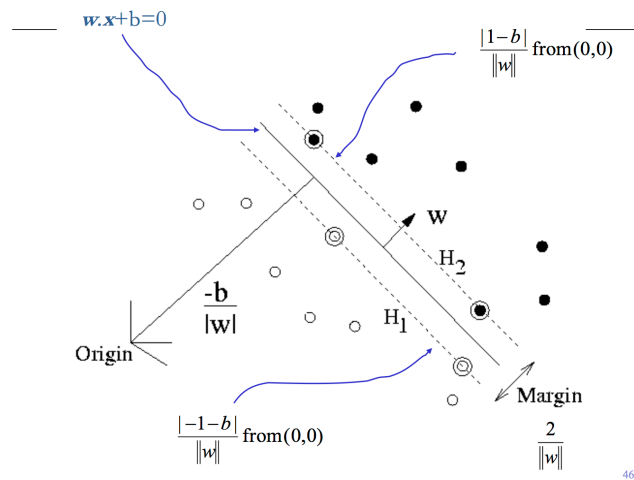
Suppose there is a separating hyperplane (i.e. linear function)  $w \cdot x + b = 0$  which separates positive from negative examples. This means that all training objects fulfill:

$$x_i \cdot w + b \geq +1 \text{ if } y_i = +1$$

$$x_i \cdot w + b \leq -1 \text{ if } y_i = -1$$

equivalently:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \forall i$$



$n$  objects :  $x_i \in \mathbb{R}^m$ ,  $i=1, \dots, n$  and their respective labels  $y_i \in \{-1, 1\}$

We introduce slack variables  $\xi_i$ :

$$\begin{aligned} x_i \cdot w + b &\geq +1 - \xi_i \text{ if } y_i = +1 \\ x_i \cdot w + b &\leq -1 + \xi_i \text{ if } y_i = -1 \end{aligned}$$

New objective function:

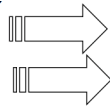
$$\|w\|^2 / 2 + \gamma (\sum \xi_i)$$

Linear SVM – Classifier

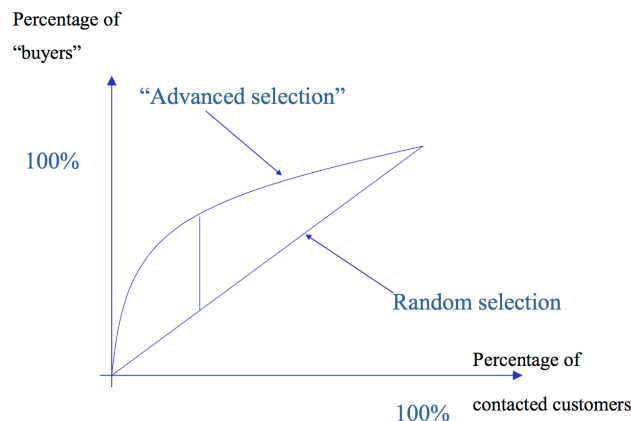
$$f(x) = W \cdot x + b = \sum_i \alpha_i y_i \cdot x + b$$

Determining  $\text{sign}(f(x))$

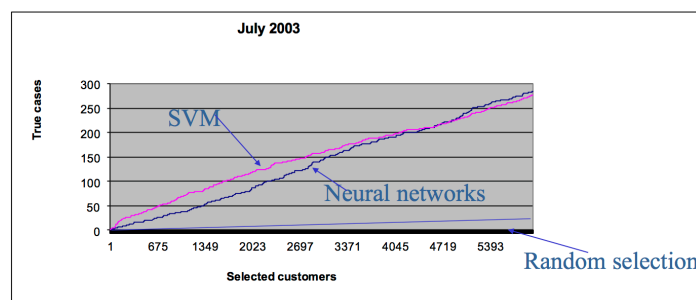
- ◆ If  $\text{sign}(f(x)) = +1$
- ◆ If  $\text{sign}(f(x)) = -1$



Belongs to class +1  
Belongs to class -1



## Comparing Neural networks, SVM, and random selection

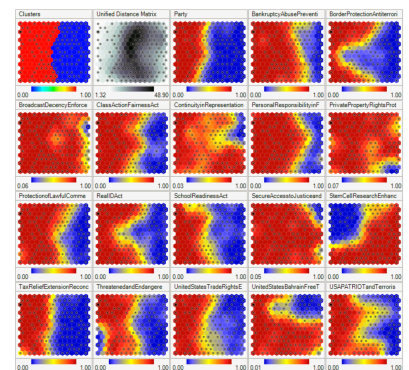


Selection: 100 => True with NN=5, True with SVM=13, Random selection=0,4  
Selection: 500 => True with NN=20, True with SVM=38, Random selection=2  
Selection: 1000 => True with NN=36, True with SVM=68, Random selection=4

## Kohonen Self Organising Maps

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction. Self-organizing maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighbourhood function to preserve the topological properties of the input space.

A self-organizing map consists of components called nodes or neurons. Associated with each node are a weight vector of the same dimension as the input data vectors, and a position in the map space. The usual arrangement of nodes is a two-dimensional regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher-dimensional input space to a lower-dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest (smallest distance metric) weight vector to the data space vector.

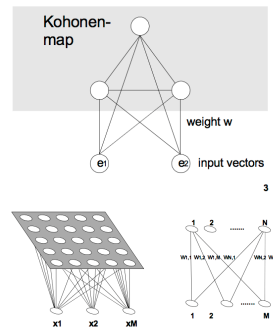


- Unsupervised learning
- Organise output neurons into array feature map (typically 1-dimensional line or 2-dimensional plane)
- All neurons in input layer are connected to all neurons in output layer
- Present training examples one by one and increase the weights of the most similar neuron ('the winner') and increase all the weights in its neighbourhood in a decreasing manner ("Winner take all learning")
- Useful for clustering and visualisation

Unsupervised learning with neural networks: Kohonen feature map

Application

- Clustering
- Visualization



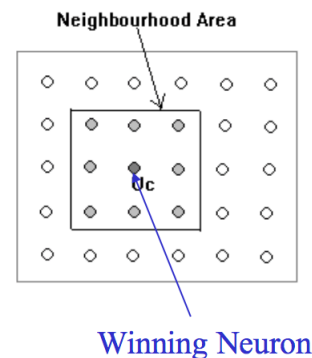
## Self-organizing feature maps

Core formula of training algorithm:

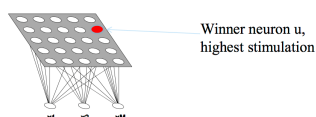
$$\mathbf{Wv}(s+1) = \mathbf{Wv}(s) + \Theta(u, v, s) \alpha(s)(\mathbf{D}(t) - \mathbf{Wv}(s))$$

With:

$\mathbf{Wv}(s)$	: Weight vector in iteration $s$
$\Theta(u, v, s)$	: Neighbour function
$\mathbf{D}(t)$	: Input vector
$\alpha(s)$	: Learning rate
$s$	: Iteration
$t$	: Input vector
$u$	: Winner neuron or BMU (best matching neuron)
$v$	: Any neuron of the SOM



## Neighbourhood in SOM



$\Theta(u, v, s)$ : "similarity" (here: excitation) between winner neuron  $u$  and any neuron  $v$  in iteration  $s$ .

Some neighbourhood functions: RBF, "Mexican hat", constant, ...





**Benefits of Neural Networks**

- Networks with 1 hidden layer are universal approximations
- Very good generalization capability (noise resistant)
- Non-parametric techniques (e.g. no normality assumptions)
- Allow to effectively deal with high dimensional, sparse input spaces

**Drawbacks of Neural Networks**

NNs are black box techniques

- No easy comprehensible relationship between inputs and outputs
- trade-off between model accuracy and model comprehensibility
- But: Techniques to extract decision rules out of trained Networks (e.g. NeuroFuzzy Systems)

How to choose network topology?

- E.g. hidden layers, hidden neurons, activation functions, ...
- But: New learning algorithms fairly robust w.r.t. network topology

Local minima

## CHAPTER 19: ENSEMBLE LEARNERS, SURVIVAL ANALYSIS, SOCIAL NETWORK ANALYTICS, ORGANIZATIONAL ASPECTS OF ANALYTICS

### 1. Ensemble Learners

In a given scenario, it may prove more useful to chain or group classifiers together, using the techniques of voting, weighting, and combination to pursue the most accurate classifier possible. Ensemble learners are classifiers which provide this functionality in a variety of ways.

Idea: ensemble learners are ways to further improve the algorithms or classifiers. The basic idea is to combine learners.

- The same algorithm is applied several times
- Different algorithms are applied

How to combine learners?

- Bagging
- Boosting
- Stacking

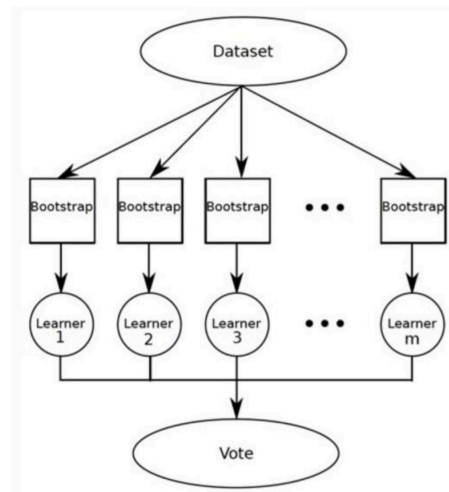
#### 1.1 Bagging

Idea: Take different Bootstrap samples and apply sampling with repetition. Basically I apply one technique on each sample. I then apply voting to choose the majority class. It is interesting for unstable learners to get better predictive models.

⇒ Bagging is applied in parallel.

Bagging operates by simple concept: build a number of models, observe the results of these models, and settle on the majority result. Ex: I recently had an issue with the rear axle assembly in my car: I wasn't sold on the diagnosis of the dealership, and so I took it to 2 other garages, both of which agreed the issue was something different than the dealership suggested. The accuracy would likely increase if I had visited tens or hundreds of garages. This holds true for bagging, and the bagged classifier often is significantly more accurate than single constituent classifiers. Also note that the type of constituent classifier used are inconsequential; the resulting model can be made up of any single classifier type.

Bagging is short for bootstrap aggregation, so named because it takes a number of samples from the dataset, with each sample set being regarded as a bootstrap sample. The results of these bootstrap samples are then aggregated.

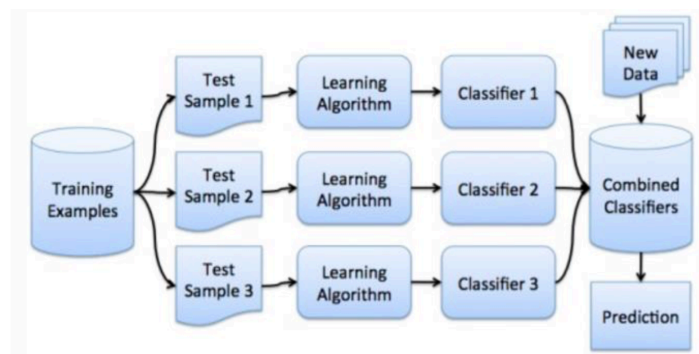


- Operates via equal weighting of models
- Settles on result using majority voting
- Employs multiple instances of same classifier for one dataset
- Builds models of smaller datasets by sampling with replacement

Works best when classifier is unstable (decision trees, for example), as this instability creates models of differing accuracy and results to draw majority from

Bagging can hurt stable model by introducing artificial variability from which to draw inaccurate conclusions

## 1.2 Boosting



Boosting is similar to bagging, but with one conceptual modification. Instead of assigning equal weighting to models, boosting assigns varying weights to classifiers, and derives its ultimate result based on weighted voting.

Ex: Thinking of the car problem, perhaps I had been to one particular garage numerous times in the past, and trusted their diagnosis slightly more than others. Also suppose that I was not a fan of previous interactions with the dealership, and that I trusted their insight less. The weights I assigned would be reflective.

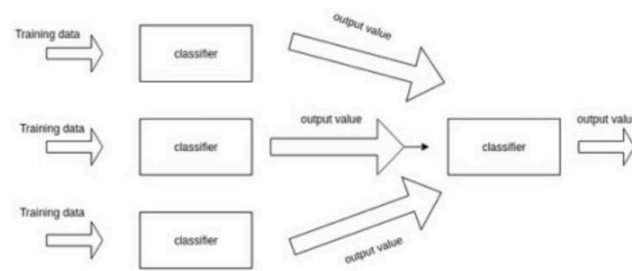
So basically, I don't treat all learners equally but I consider some learners as more important than others;

⇒ This is a sequential process, so the now model is influenced by the previous model. Focus on misclassified observations.

Ex: AdaBoost is a popular Boosting Algorithm. LogitBoost (derived from AdaBoost) is another, which uses additive logistic regression, and handles multi-class problems.

- Operates via weighted voting
- Algorithm proceeds iteratively; new models are influenced by previous ones
- New models become experts for instances classified incorrectly by earlier models
- Can be used without weights by using resampling, with probability determined by weights
- Works well if classifiers are not too complex
- Also works well with weak learners

### 1.3 Stacking



Stacking is a bit different from the previous 2 techniques as it trains multiple single classifiers, as opposed to various incarnations of the same learner. While bagging and boosting would use numerous models built using various instances of the same classification algorithm (eg. decision tree), stacking builds its models using different classification algorithms (perhaps decision trees, logistic regression, an ANNs, or some other combination).

A combiner algorithm is then trained to make ultimate predictions using the predictions of other algorithms. This combiner can be any ensemble technique, but logistic regression is often found to be an adequate and simple algorithm to perform this combining. Along with classification, stacking can also be employed in unsupervised learning tasks such as density estimation.

⇒ Creates better classifiers

### 1.4 Random Forest

Idea: random forests builds several decision trees based on different sets of input variables, randomly selected to reduce correlation. It combines models by means of bagging.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Advantages:

- Well-performing
- Robust to outliers and noise
- Parallelizable

⇒ What are the advantages of ensemble learners?

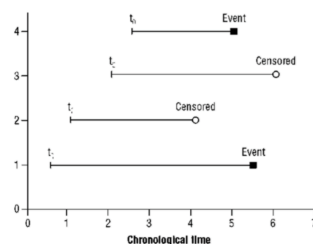
- Ability to turn weak classifier into strong classifier by aggregation
- Bagging works well when high variance by reducing overfit
- Boosting works well when high bias since it will focus on misclassifications of previous models
- Ensembles are well performing and interesting to use, but it may make it more difficult to interpret our results. Next, the computation time will increase.

⇒ What about the understandability and computation time?

## 2. Survival Analysis

Survival analysis is useful to predict time until an event. Ex: how long until the customer churns, how long until a customer defaults, time until failure, etc.

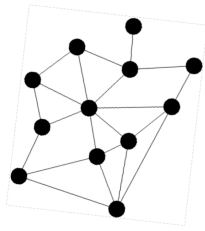
Analysed data in survival analysis is very special as it is right-censored data. We know the start time but we don't always know the end time, we ignore what happened to our data.



It is mainly composed out of **2 elements**

- **Survival function:** how large is the chance we will survive until time  $t$ ? the mean is the expected survival time.  $St = 1 - Ft = 1 - P(T \leq t)$
- **Hazard function:** expected survival rate. Ex: what is the probability that a certain customer will churn given that he hasn't churned yet.

## 3. Social Network analytics



Social network analysis is the analysis of social structures using networks and graph theory. Social networks do exist all around us, think of Facebook, Twitter, Friends, ... Call networks are another example. They are generated based on who we call or text.

### 3.1 Networks explained

Networks consist of

- Nodes or Vertices
  - Real life entities
  - People, companies, etc.

Ex: represent people we connect

Nodes can be **labelled** based on the class they belong to: churning, non-churning, ... Use these labels to predict the outcome of someone we don't know its label of.

- Edges or links:
  - Connections between the nodes
  - Friendships, relationships, cooperation

Edges can be **directed** or **undirected**. Ex: on Instagram connections are directed. I follow some people but these people do not necessarily follow me. On the contrary, Facebook includes undirected connections.

Edges can also be **weight** by putting a number representing how strong the connection is, it represents the intensity of the relationship. Ex: number of phone calls between people.

In real life applications, information from the network is used to predict the labels of the nodes.

- **Featurization**: for each node we can extract some interesting measures. In other words, measure the impact of social environment on the nodes of interest
  - Degree: size of neighbourhood. Count the number of connections
  - Density: how well is my neighbourhood connected?
  - Closeness: mean distance from a node to each other node in the network
  - Betweenness: number of shortest paths going through a node, it indicates how much information flows through the node.

Degree and density are both neighbour measures whereas closeness and betweenness are centrality measures as they measure the connectedness within the whole network not just with few nodes. However, it is not always scalable to work with these two measures as they can't handle well with large networks.

After computing these different measures, we end up with a flat dataset, a table providing us with interesting measures.

CustomerID	Degree	Density	Betweenness	...	Churn
1	6	0.4	104		Yes
2	1	0.1	13.83		No
3	4	0.11	9		No
4	88	0.67	74		Yes
5	2	0.4	0		No
...	...	...	...	...	...
N	3	0.9	12		No

- Extract various features for each node in the network
- The network represented in a table
- Use the feature sin a predictive framework, such as churns

- **Network learning**

### 3.2 Homophily explained

⇒ Are there observable effects present in the network?

- Expert knowledge: relationships between churners
- Confirmed by descriptive network analysis
- Homophily in social networks (from sociology): people have a strong tendency to associate with other whom they perceive as being similar to themselves in some way. Ex: same city, hobbies, interests, ...
- Homophily in churn networks: churners are more likely to be connected to other churners, and non-churners are more likely to be connected to other non-churners.

⇒ Networks does contain statistically significant patterns of homophily.

**1. Cross-labeled edges:**

Number of observed cross-labeled edges should be significantly less than number of expected cross-labeled edges

$$H_0: \hat{r} \leq 2 \cdot p_{nf} \cdot p_f$$

**2. Dyadicity:**

Extent to which churn nodes are connected to churn nodes

$$D = \frac{m_{11}}{\bar{m}_{11}}$$

**3. Heterophilicity:**

Extent to which churn nodes are connected to churn nodes

$$H = \frac{m_{10}}{\bar{m}_{10}}$$

### 3.3 Case study: Churn prediction in telco

Churn in Telco is a hot topic. It consists of predicting churn in Telco, based on logs. Logs could for instance give some information about calls: who called who, when, etc. The main purpose is to build a network to catch interesting information.

Today, the Telco market seems quite saturated, hence it is important for companies to keep their customers and make sure they don't go to competitors.

Telco's have to deal with a continuous flow of data to build these networks.

- Customer churn prediction is **important** in telco
  - Attrition is more expensive than retention
  - Easy for customers to change providers
  - Happy customers with long relationships will spread the word
- Customer churn prediction is **pertinent** in telco
  - A lot of various types of data
  - Fast moving market
- Classical way of CCP
  - Use dataset with customer variables
  - Use binary classifiers: Logistic Regression, Random Forest, Neural Network, etc.
- Enriching the dataset with network variables (Featurization) increases model performance

Predict churn from networks directly using Relational Learners

- Simulate how 'churn influence' spreads through the network using
  - The strength of ties between customers in the networks
  - Information about who has and hasn't churned

to infer churn probabilities

Relational Learners are composed of

1. Relational Classifiers: Infer labels for each node, based on links to neighbouring nodes and their labels
2. Collective Inference methods
  - Regulate how nodes are labelled together and in which order
  - Make multiple, subsequent inferences, making the inferences more stable
  - Determine a final label or score

## 4. Organizational aspect of analytics

### 4.1 How to organize your data science team?

How to organize my team? Companies are valuing data and analytics a lot, but it requires a lot of change in decision and the way of working.

*"There may be no single right answer to how to organize your analysts- but there are many wrong ones".*

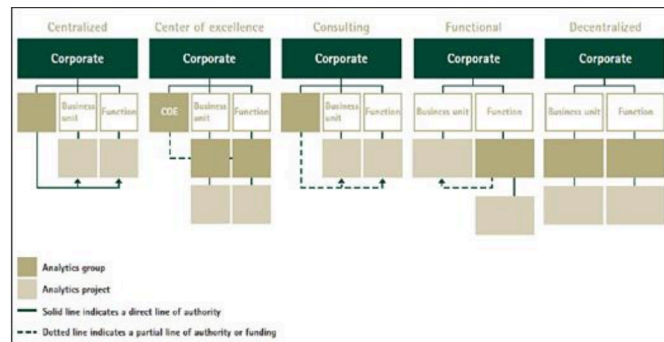


DELTA framework (Davenport et al., 2010)



Five categories are used to assess the level of maturity. How to move from one level to the next one?

- **Data:** storing data, data quality, how to integrate data from different sources, countries, etc. how to access dat. If bad data, we can't trust our results.
- **Enterprise:** how are you looking at analytics in the enterprise? Ad hoc? Departmental? Lot to do with culture. How are analytics propagated through the organization?
- **Leadership:** who is managing analytics at a higher level?
- **Target:** at which level do you set targets for analytics? High level = strategic level. It describes the way the organization works.
- **Analysts:** way to organize depends on our competitors, time, ... Five models are used in this category
  - **Centralized model:** organize data scientist at corporate level. There is one corporate organization for analytics.
    - Advantages: repeatability, scalability
    - Drawbacks: distance business, analytics
  - **Consulting model:** Business units "hire" data scientists as consultants to analytical projects.
    - Advantages: market driven, advisory role
    - Drawbacks: priority to units with high budget
  - **Functional model:** Single data science team in each department. Data scientists within a business department but can easily rotate, taking knowledge and skills with them.
    - Advantages: migration of data scientists, close to business
    - Drawbacks: level of engagement
  - **Centre of excellence:** Decentralized data scientists, but member of CoE. data scientists are spread among the company but still combined into a team.
    - Advantages: community, education
  - **Decentralized model:** Nor corporate or consolidating organization. Most occurring mode, least mature way.
    - Advantages: can be effective in diversified organizations
    - Drawbacks: difficult to set priorities, education, resources, standardization



A lot of companies are looking for data scientist, but there are not enough people trained. There is a growing shortage of data scientists  $\Leftrightarrow$  rise of the “citizen data scientist”. There is a need for empowering novice, business users in applying analytics.

How? **Analytics-as-a-Service:**

- Cloud-based
- Usage-based pricing
- Easy-to-use
- Fast development & deployment of analytics models
- Flexibility of resources
- Scalability
- But what about privacy, security, accountability

$\Rightarrow$  Vendors have been developing solutions to empower business users

#### 4.2 Experimental study

- BigML: if you upload data you can easily visualize it, and apply existing techniques
- Azure: easy to use and make models within companies, across communities, etc.
- RapidMiner