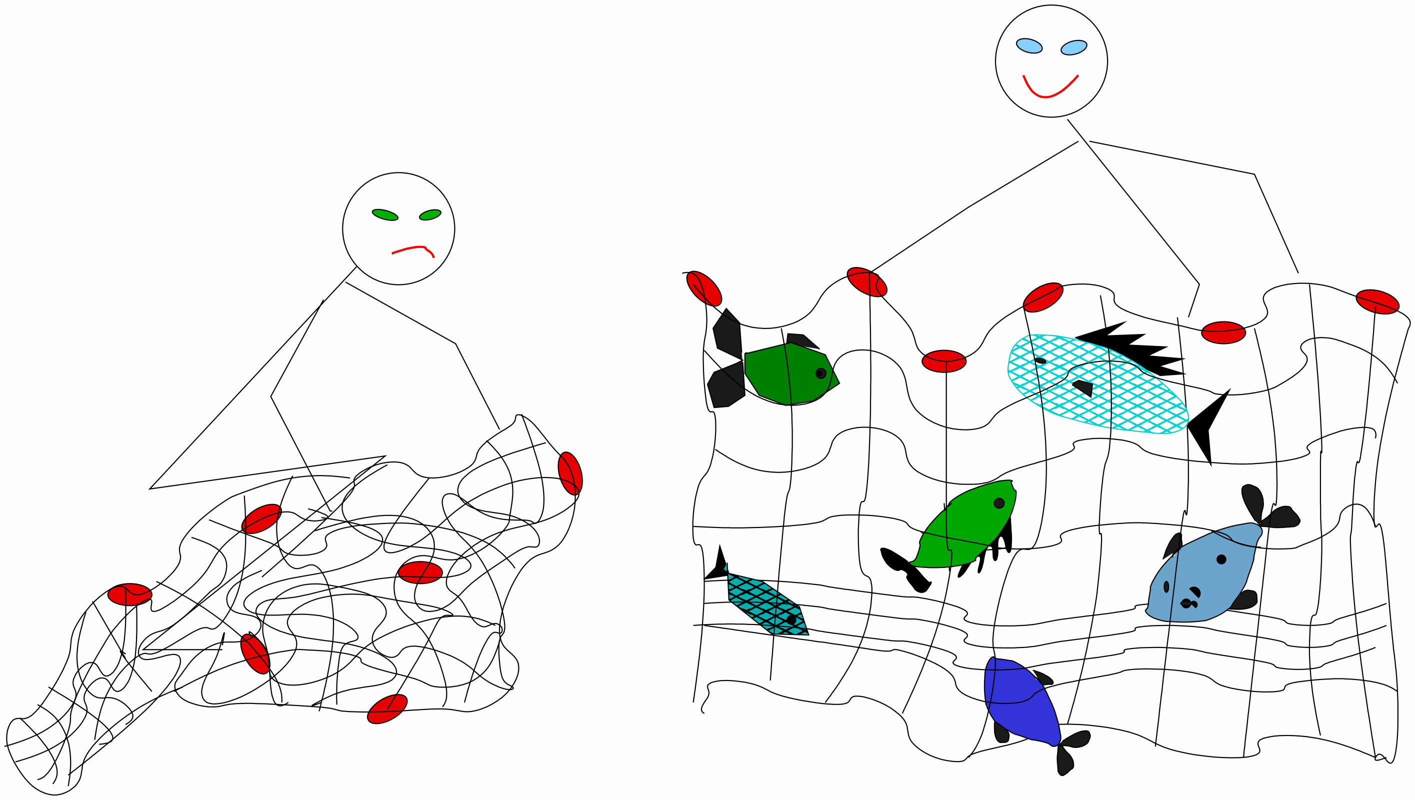
CS-E4650 Methods of Data Mining



I Course logistics, II Introduction to DM, III Preprocessing

Teaching staff

instructor: Wilhelmiina “**Wiki**” Hämäläinen (wilhelmiina.hamalainen@aalto.fi)

teaching assistants:
Egor Eremin egor.eremin@aalto.fi
Georgy Ananov georgy.ananov@aalto.fi
Hieu Nguen Khac hieu.nguyenkhac@aalto
Lai Khoa khoa.lai@aalto.fi
Paavo Reinikka paavo.reinikka@aalto.fi
Yinjia Zhang yinjia.zhang@aalto.fi
+ Vinh N’guyen helped with preparation

guest lecturers or visitors:
prof. Heikki Mannila and MSc Juho Rinta-Paavola

contact: course forum, please avoid email chaos!

Communication and course material

All course information available via **mycourses.aalto.fi** (MC):
`https://mycourses.aalto.fi/course/view.php?id=41020`

- announcements (all important announcements by MC!)
- lecture notes and external material
- link to the text book: **Charu C. Aggarwal: Data Mining: The Textbook**, Springer 2015
- exercise tasks and material
- link to **course forum** `https://mdm2023.zulip.aalto.fi/`

Ask during lectures and exercise sessions and in the course forum. **Please, use email only for personal matters** that you cannot ask elsewhere.

Advicing in zulip

Questions on exercises and homeworks

- ask under the right channel (e.g., “Exercise session 1”)
- give informative title to the stream (like task number)
- TAs’ reply questions during weekdays (+ other students can reply)
- no real-time responses (some delays)
- if you want a reply before weekend, ask before Thu 4pm latest

Other questions (lectures, general)

- like above, but the lecturer and TAs reply (also students can reply, if you know the answer! e.g., something told in MC)

Completing course

1. activite participation in exercise groups (5 sessions, max 5p)
2. submitting homeworks in groups of 2–3 students (5 tasks, max 10p)
3. final **exam** Wed 13.12. 13:00–16:00 (max 24p)
4. prerequisite test (max 1p) `https://plus.cs.aalto.fi/cs-e4650/2023/` (**deadline 18th Sep 2023**)

- the final grade is based on the sum of points (max 40)
- to pass the course one needs to get $\geq$ **50% of total points and** $\geq$ **50% of the exam points**

Exercises and homeworks

Exercise tasks

- individual solution beforehand
- processing in small groups during sessions + presentation
- in exceptional/force majeure circumstances you can once return a solution report to the TAs instead

Homeworks (home assignments)

- done in groups of 2–3 students (but independent work, no AI tools unless specifically asked to use)
- at least 10 days time to solve
- submit before the deadline! (with -10% penalty can be 24h late)

Average workload (5 ects $\approx$ 135h)

- 34–36h [a] contact sessions (lectures and exercises)
- 20h preparation for exercises
- 20h graded homeworks (in groups)
- 40h self-studying (more if skipped lectures/sessions)
- 20h preparation for the exam

Important: Solve exercise tasks beforehand! (Best way to learn!)
Self-study every week! (read the book & other learning material)

[a] now allocated 1 extra lecture

Learning goals

- Know fundamental data mining problems, pattern types and methods
- Know which methods to choose for a given problem or keywords to find more information
- Recognize when to expect computational problems and know some feasible strategies
- Understand importance of validation and know some approaches to validation
- Make programs that use or implement DM methods
- Utilize existing source code and tools in DM tasks
- Learn good DM practices

Meta-learning goals

Not actual learning goals, but **useful skills for data miners** that you are encouraged to learn!

- reading scientific papers related to DM
- writing efficient programs (and algorithms)
- managing many alternative tools or programming languages
- working in linux/unix environment
- learning critical thinking

https://medium.com/codex/how–slow–is–python–compared–to–c–3795071ce82a

Syllabus

- Introduction to DM
- Data preprocessing
- Distance and similarity
- Clustering (extensions of K-family, hierarchical, spectral + evaluation)
- Association mining
- Graph mining
- Social network analysis
- Web mining and recommendation systems
- Text mining
- guests: Data randomization, Episode mining

Relationship to some other courses

pattern discovery

some unsupervised ML

Methods of data mining

Machine learning

introduction to
both supervised and
unsupervised ML

clustering, regression,
classification, ...

Machine learning:
Supervised methods

classification, linear models, ...

Special courses, e.g.,

Bayesian data analysis

ML: Advanced probabilistic methods

Complex networks

Research project in ML, DSC and AI

ML for bioinformatics

Statistical NLP

Kernel methods in ML

Deep learning

Prerequisites: Important!

1. Basic mathematics and statistics

- reading mathematical notations
- basic concepts of probability theory (distributions, conditional probability, independence, probability calculus)
- basic concepts of statistics (summary statistics like mean, median, variance, covariance, idea of statistical significance)
- basic matrix algebra (basic operations, some notion of eigenvalues and eigenvectors)

Prerequisites (cont'd)

2. Programming

- ability to process data and implement algorithms in some well-known programming language (Python, Java, C, C++, Matlab)

3. Algorithms and data structures

- reading pseudocode
- lists, trees, graphs etc.
- O-notation, NP-hardness
- basic algorithm strategies

Ask if you don't know something!

- utilize the **course forum**! It is most efficient!
  - channels for general/practical things, lectures and material, exercises, assigments
  - check extra clarifications, what others have asked and ask new questions
- ask during lectures
  `https://presemo.aalto.fi/mdm2023`
- take advantage of the exercise sessions
- read the textbook and extra materials
- make study groups with your colleagues
- use library and internet

Introduction to Data Mining

- What is data mining?
- Data mining process

Image sources: https://www.pikist.com/
https://howtofindgoldnuggets.com/metal–detect–fringe–spots–find–gold–nuggets/

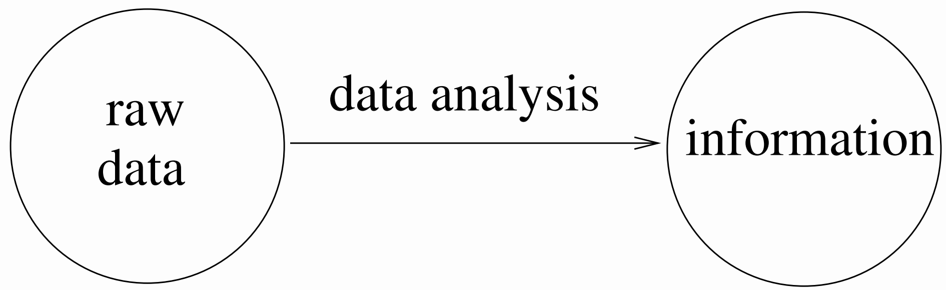
What is Data mining (DM)?

- no definite and clear answer
- computationally nontrivial data analysis for finding new useful **information** from large collections of **data**
  - interesting patterns like relationships and groupings
- Challenge: data volumes are all the time inreasing!

  $\Rightarrow$ more efficient algorithms needed

  $\Rightarrow$ number of patterns and spurious discoveries increases $\Rightarrow$ How to find interesting and reliable patterns?

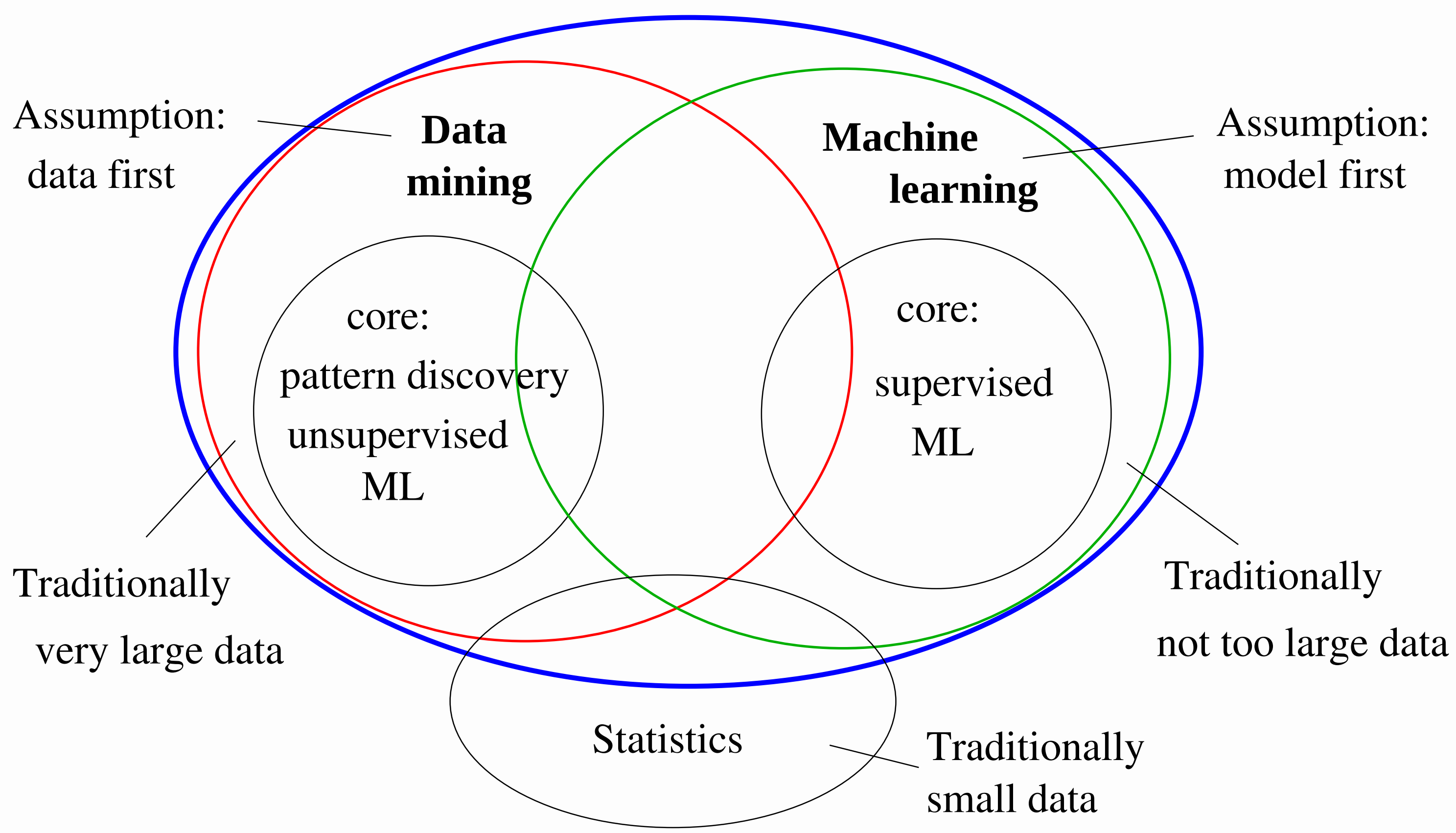
Data vs. Information?



- raw data = unprocessed, uninterpreted facts (e.g, measurements)
- information = knoweldge that has meaning, “interpreted data”
- relative terms: the resulting information from one process may be source data for another process

Relationship to closest neighbouring fields

DM ~ **knowledge discovery (from databases)** (KDD)
Machine learning strongly overlapping/synonymous!



Model vs. pattern?

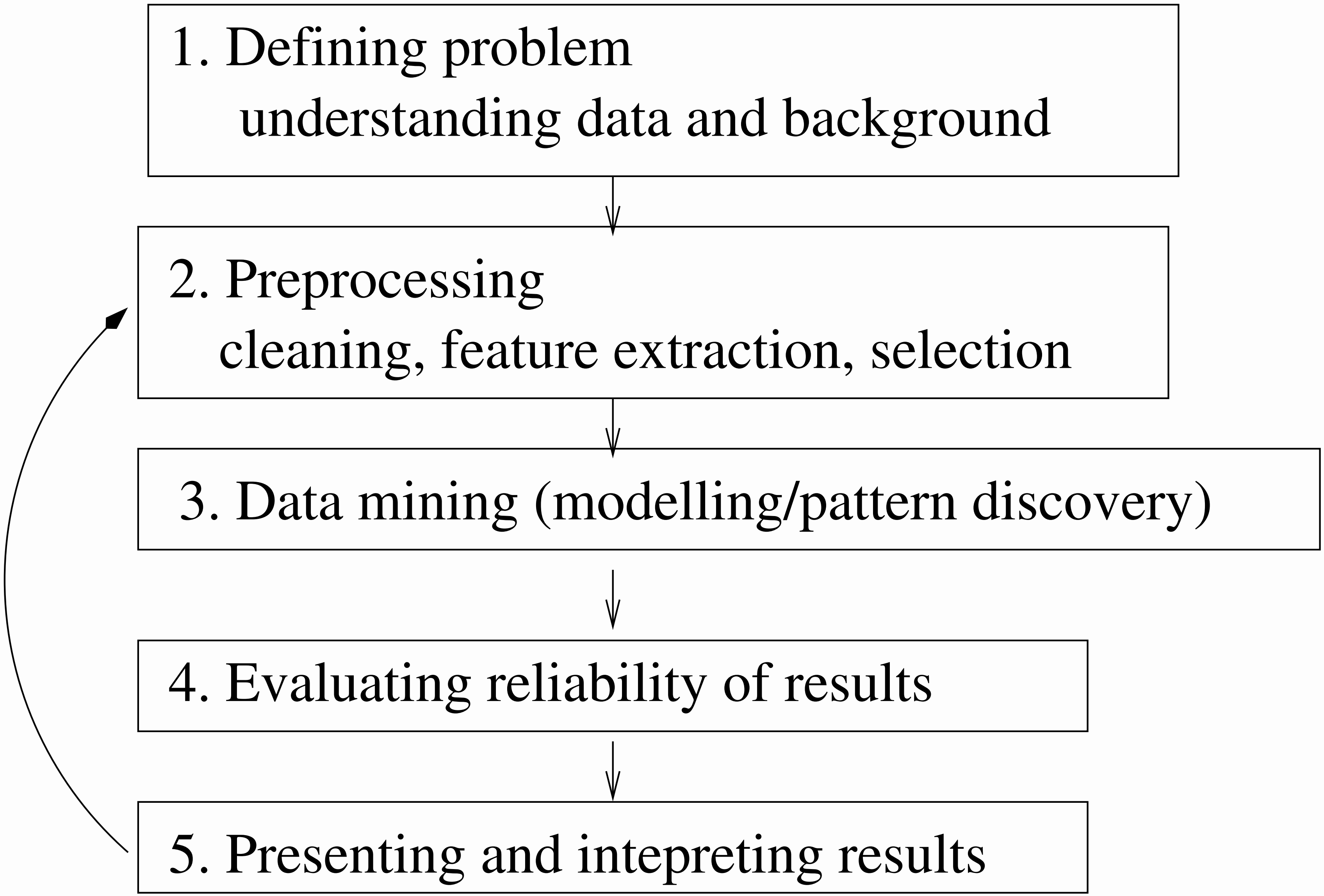
Model

- global (fits entire data)
- e.g., course success (passing the course) can be predicted from exercise points, time spent on course and participation in exercise groups

Pattern

- local model (describes some part of data)
- e.g., if students obtain high points in assignment 2 they tend to obtain high points also in the exam task 3

DM process


1. Defining problem
understanding data and background
2. Preprocessing
cleaning, feature extraction, selection
3. Data mining (modelling/pattern discovery)
4. Evaluating reliability of results
5. Presenting and intepreting results

1. Defining the problem

- Understanding data: what variables measure/describe?
- What are data types? How much there is data?
- What kinds of patterns would be interesting or useful to find?
- What is already known?
- It is worth studying some background theory!
- Difficulty: How people from different fields find the same language?

Example: defining problem

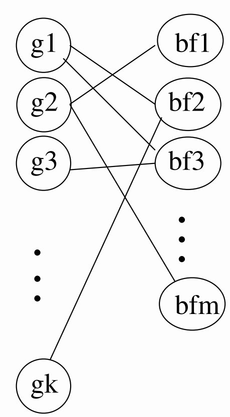
Medical scientist: How TNF-α stimulation affects gene regulation in prostate cancer cells and which biological functions are involved?

Computer scientist: First, explain the data

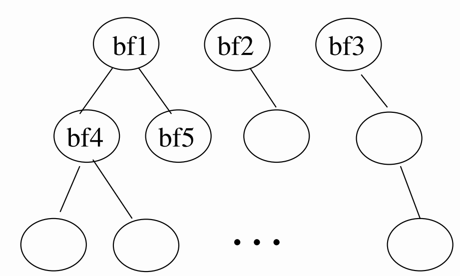
Data matrix: expression of genes g1,...,gk and class

id	expression values of g1	g2	g3	. . .	gk	class
s1	7.1	2.3	4.6		3.1	cancer
s2	6.5	8.0	4.9		5.4	healthy
s3						cancer
				⋮		healthy
						healthy
						cancer

Biological functions of genes



Ontology of biological functions



So, I should find g_is that differ significantly in two groups and corresponding bfs?

2. Preprocessing

- Combining data from different sources (may require transformations)
- Preliminary analysis: means, standard deviations and distributions of variables, correlations, ... (e.g., with statistical tools)
- Data cleaning: handling missing values, detecting and correcting errors
- Feature selection and extraction
- Possibly dimension reduction (combines feature extraction and selection)

3. Data mining

- Typical building blocks dependency analysis, classification, clustering, outlier detection
- Always good to begin from dependency analysis! → choosing features and modelling methods
- Usually descriptive modelling helps in building a predictive model
  - e.g., gene–habit–disease data
  - Descriptive: Find 100 most significant association rules related to variable Diabetes
  - Predictive: Learn (from selected data) the best model that predicts diabetes

4. Evaluating reliability of results

- Are discovered patterns or models sensible?
  - it is possible there are no models or patterns in the data – but **the methods tend to return something even from random data!**
- validating predictive models easy (test set, cross validation)
- evaluating reliability of descriptive models more difficult
- Goal: Some guarantees that the **discovered pattern is not due to chance**
- tools: statistical significance testing, use of validation data

5. Presenting and interpreting results

- present results illustratively so that essential things are emphasized
- domain experts have an important role!
- Did you find something new? Could you formulate a hypothesis based on results? What should be studied further?
- leads often to a new DM round; try new variables and possibly other methods
- finish the iterative DM process when you are satisfied or nothing new seems to be discovered

III Data types

Many ways to characterize data types

- structured or unstructured
- dependency-oriented or nondependency-oriented
- numerical, categorical or mixed
- static $\leftrightarrow$ temporal; spatial; spatio-temporal

Structured vs. Unstructured

- **Structured**
  - has a predefined structure (e.g., rows and features)
  - e.g., multidimensional, graph-formed, time series
- **Unstructured**
  - no pre-defined format, just a string
  - e.g., text, audio, video, signal data
- **Semistructured**
  - contains internal tags that identify separate data elements
  - e.g., XML documents, emails

Dependency-orientation

- Nondependency-oriented: no specified dependencies between objects or attributes
- Dependency-oriented: data objects or values related temporally, spatially or through network links
  1. **explicit dependencies**
     - relationships in graph or network data
  2. **implicit dependencies**
     - known to typically occur
     - e.g., consecutive temperature readings likely similar

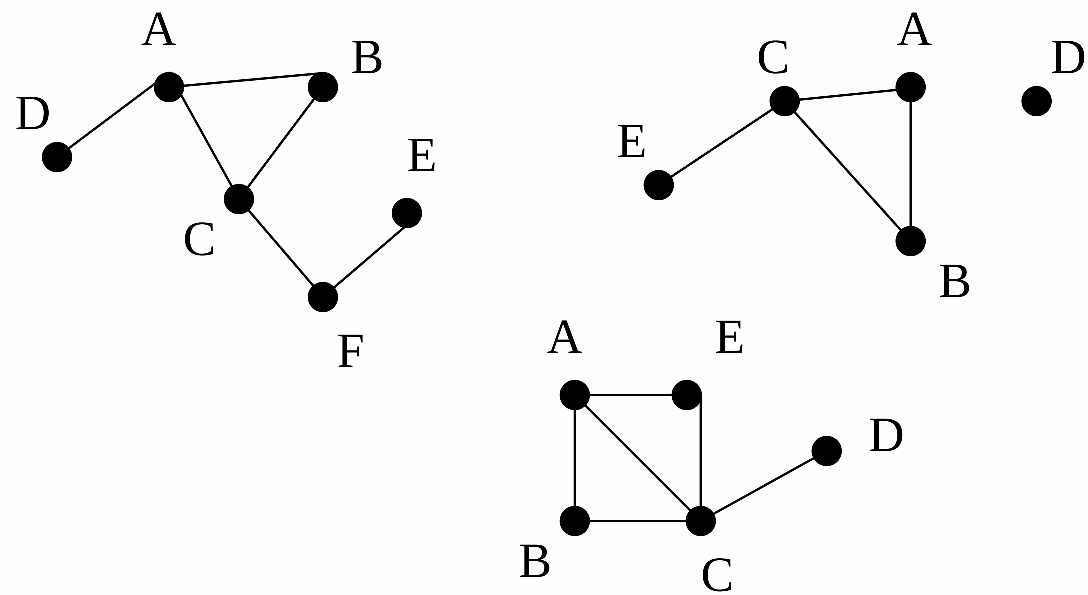
Difference: dependencies in data type vs. patterns in data instances

DATA TYPE ⟷ DATA INSTANCES

graph

Dependencies in data structure:

edges present relationships



Discovered dependency: clique of A, B and C

occurs frequently

Implicit dependencies harder to separate from patterns!

Basic data type: Multidimensional data

- a set of records, whose fields are features
- notate $\mathcal{D} = \{\overline{X_1}, \ldots, \overline{X_n}\}$, where $\overline{X_i} = (x_i^1, \ldots, x_i^d)$
  - n rows (records, data points, instances, objects) and d features (fields, attributes, dimensions)
- suitable for a relational database, e.g., cow data:

name	race	weight	parity	milk/d	activity
Rose	Holstein	640	2	35	4800
Daisy	Ayrshire	675	3	37	5100
Strawberry	Finncattle	615	4	28	7200
Molly	Ayrshire	650	1	32	6300

Numerical, categorical or mixed?

Depending on the type of variables, data may be called numerical (quantitative), categorical or mixed (both).

Variables can be classified by measurement scales:

1 Categorical

1.1 **Nominal**: values are only labels, **no order**
- e.g., gender (binary), colour, home city, occupation
- mode (most common value) is defined

1.2 **Ordinal**: values have an **order**
- e.g., satisfaction with services: very unsatisfied, unsatisfied, neutral, satisfied, very satisfied
- mode and median (the middle value) defined

Measurement scales (cont'd)

2 Numerical

2.1 **Interval scale**: difference between values is defined, but **not ratio**

- no true zero point
- temperature 20°C is not twice as warm as 10°C!
- mean and standard deviation defined

2.2 **Ratio scale**: also **ratio** is defined

- absolute zero = absence of the measured property
- temperature in Kelvins, length, weight, duration
- mean, standard deviation, geometric mean ($(\prod x_i)^{1/n}$), coefficient of variation (σ/μ) defined

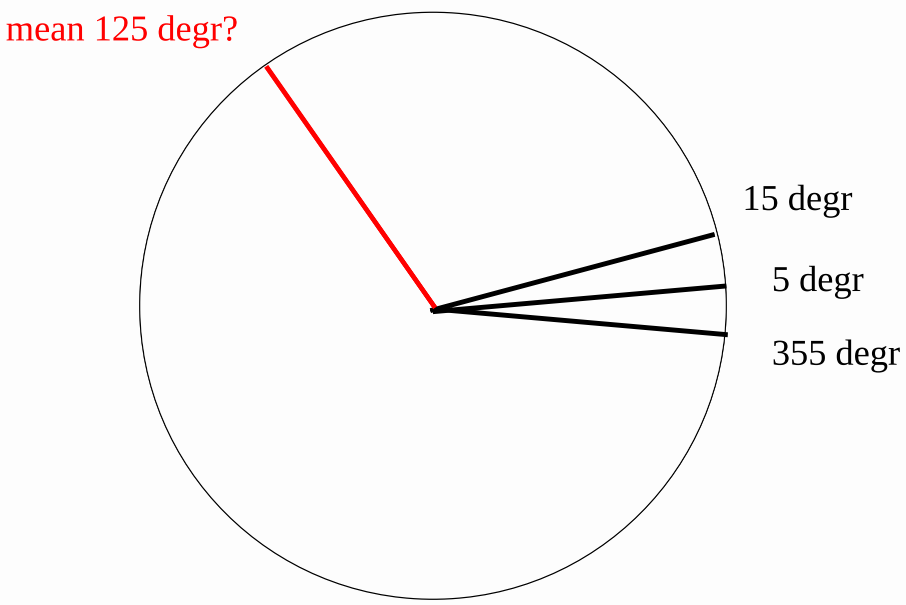
Circular variables

Idea: Values are ordered categories, where the last category precedes the first

1. Interval circular
   - e.g., compass direction (angles), time of day, day of year
   - zero on the measurement scale not meaningful!
2. Ordinal circular
   - e.g., days of the week (Mon, Tue,...), compass aspect (N, NE, E,...)

Be careful! E.g, cannot calculate arithmetic mean or normal correlation.

Example: What is the mean angle??



NO! Real mean angle 5 degr!

- present angles α_i by $(cos(\alpha_i), sin(\alpha_i))$
- $S = \sum_i sin(\alpha_i)$, $C = \sum_i cos(\alpha_i)$
- $\theta = arctan\left(\frac{S}{C}\right)$, if $S \geq 0$, $C > 0$
- $\theta = arctan\left(\frac{S}{C}\right) + \pi$, if $C < 0$
- $\theta = arctan\left(\frac{S}{C}\right) + 2\pi$, if $S < 0$, $C \leq 0$
- $\theta = \pi/2$, if $S > 0$, $C = 0$
- undefined, if $S = 0$, $C = 0$

Present other circular variables first as angles (e.g., $\alpha = \frac{h*2\pi}{24}$)

Warning: Number codes ≠ numerical variables

Categorical values have often arbitrary numerical codes that can't be interpreted as numbers!

Gender: 1 = Female, 2 = Male

Cow's race: 0 = Holstein, 1 = Ayrshire, 2 = Finncattle

- cannot measure distance or ratio or calculate mean or Pearson correlation
- you can get numerical presentation by creating dummy (binary indicator) variables for each value
  - e.g., $I_{Holstein}$=1, if race=Holstein, and 0 otherwise

Warning (cont'd)

The same holds for ordinal variables:

Opinion: 1 = fully disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 = fully agree

- if fully ordinal and distances between categories equal, variable may be treated as numerical (but not always optimal)
- more typical when many categories (≥ 7)
- Be careful!

Opinion: 0 = Don't know, 1 = fully disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 = fully agree

Other data types

- time series
- discrete sequences
- spatial data
- network and graph data
- text

Time series

- continuous measurements over time
- e.g., from environmental sensors, health monitoring devices, ECG
- at time stamps $t_1, \ldots, t_n$ measurements $(Y_1, \ldots, Y_n)$
- may also be multivariate time series $(\overline{Y_1}, \ldots, \overline{Y_n})$, where $\overline{Y_i} = (y_i^1, \ldots, y_i^d)$
- e.g., heart rate, oxygen saturation, diastolic and systolic blood pressure at every minute
- often temporal correlations (like dependencies between consecutive values or periodic patterns)

Discrete sequences

- like time series, but sequences of categorical variables
- special case: strings (no time stamps, but positions)
- e.g., event logs, strings of nucleotides (DNA, genes)

Event ID	Class	Type	Severity	Date/Time	Description
958	Audit	Log	minor	Fri Apr 23 15:03:30 2010	root : Open Session : object = /session/type : value = www : success
957	Fault	Fault	critical	Fri Apr 23 13:02:41 2010	Fault detected at time = Fri Apr 23 13:02:41 2010. The suspect component: /SYS/BL3/NET1 has fault.io.pciex.fabric.fatal with probability=50. Refer to http://www.sun.com/msg/SPX86-8001-95 for details.
956	Fault	Fault	critical	Fri Apr 23 13:02:41 2010	Fault detected at time = Fri Apr 23 13:02:41 2010. The suspect component: /SYS/BL3/NET0 has fault.io.pciex.fabric.fatal with probability=50. Refer to http://www.sun.com/msg/SPX86-8001-95 for details.
955	IPMI	Log	critical	Fri Apr 23 13:02:38 2010	ID = 1d1 : 04/23/2010 : 13:02:38 : Critical Interrupt : BIOS : PCI SERR: IOH 3 ESI
954	IPMI	Log	critical	Fri Apr 23 13:02:38 2010	ID = 1d0 : 04/23/2010 : 13:02:38 : Critical Interrupt : BIOS : PCI SERR: IOH 2 ESI
953	IPMI	Log	critical	Fri Apr 23 13:02:38 2010	ID = 1cf : 04/23/2010 : 13:02:38 : Critical Interrupt : BIOS : PCI SERR: IOH 1 ESI

Figure from https://docs.oracle.com/cd/E19140-01/html/821-0796/gjfwa.html

Difficulty: how to combine temporal data when the measuring frequency varies?

Example from a cow-house:

- body temperature and rumen acidity are measured every minute
- activity device records average activity every 15 min
- milk production (amount, protein and fat contents etc.) is measured daily
- feeding automaton event log contains time stamp, automaton id, cow id, feed type, amount and duration for every visit
- drinking automaton event log contains time stamp, cow id, amount of water and duration

Spatial and spatiotemporal data

- spatial: measurements of non-spatial attributes in spatial locations (typically 2D)
  - e.g. sea surface temperature
- spatiotemporal data
  - e.g., temperature over time or ship trajectories

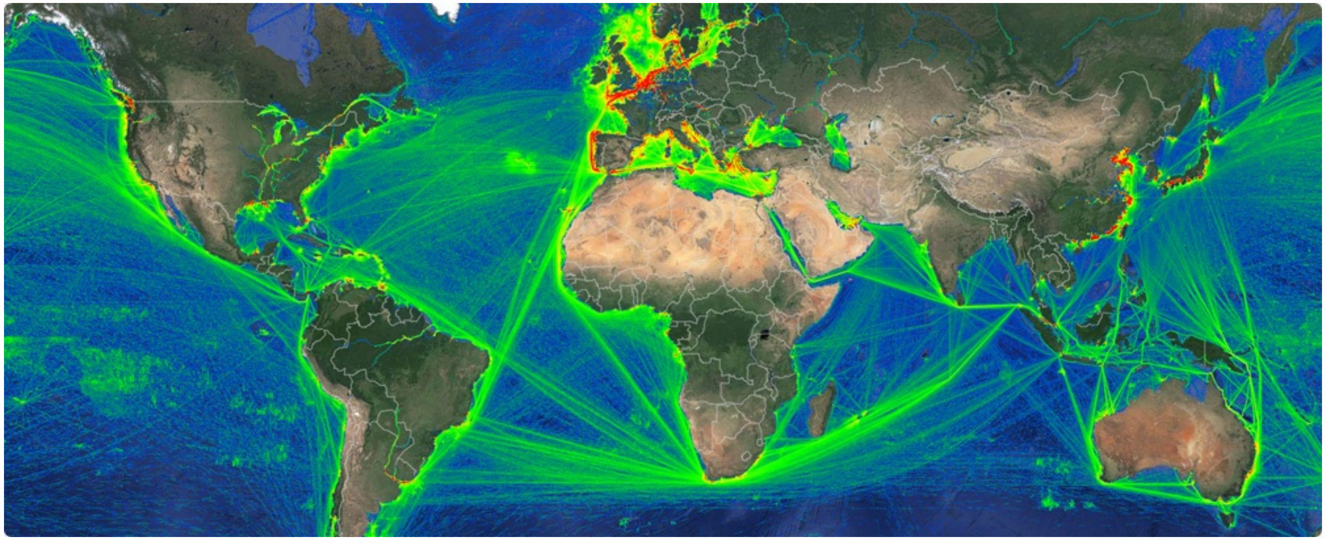


Figure from `http://www.elane.com/EN/Detail106.html`

Spatiotemporal data: contextual and behavioural attributes

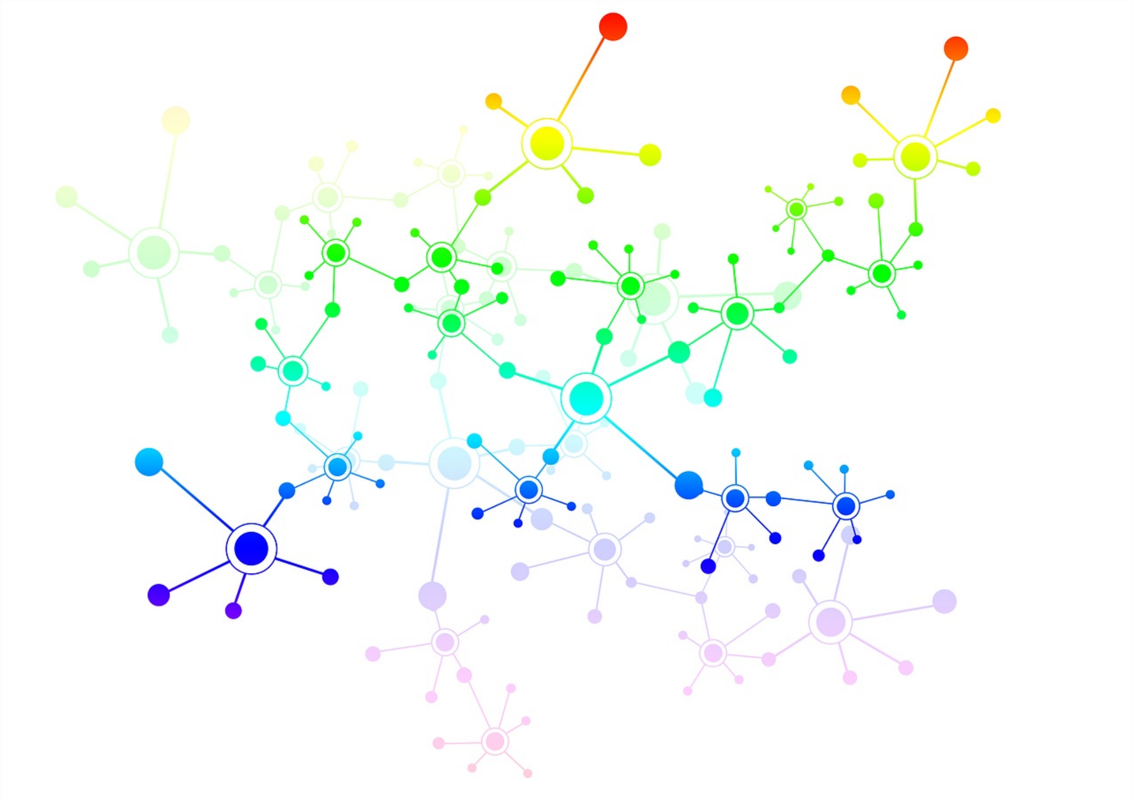
Contextual attributes define the context
Behavioural attributes are measured in this context

Two main types of spatiotemporal data:

1. Both spatial and temporal attributes define the context where some behavioural attribute (like temperature) is measured
2. Temporal attribute is contextual and spatial attributes are behavioural (e.g., trajectory analysis)

Network and graph data

- nodes correspond objects and edges relationships
  + attributes may be associated with nodes or edges
- directed (web structure) or undirected (social network)



Example: wikipedia hyperlink structure

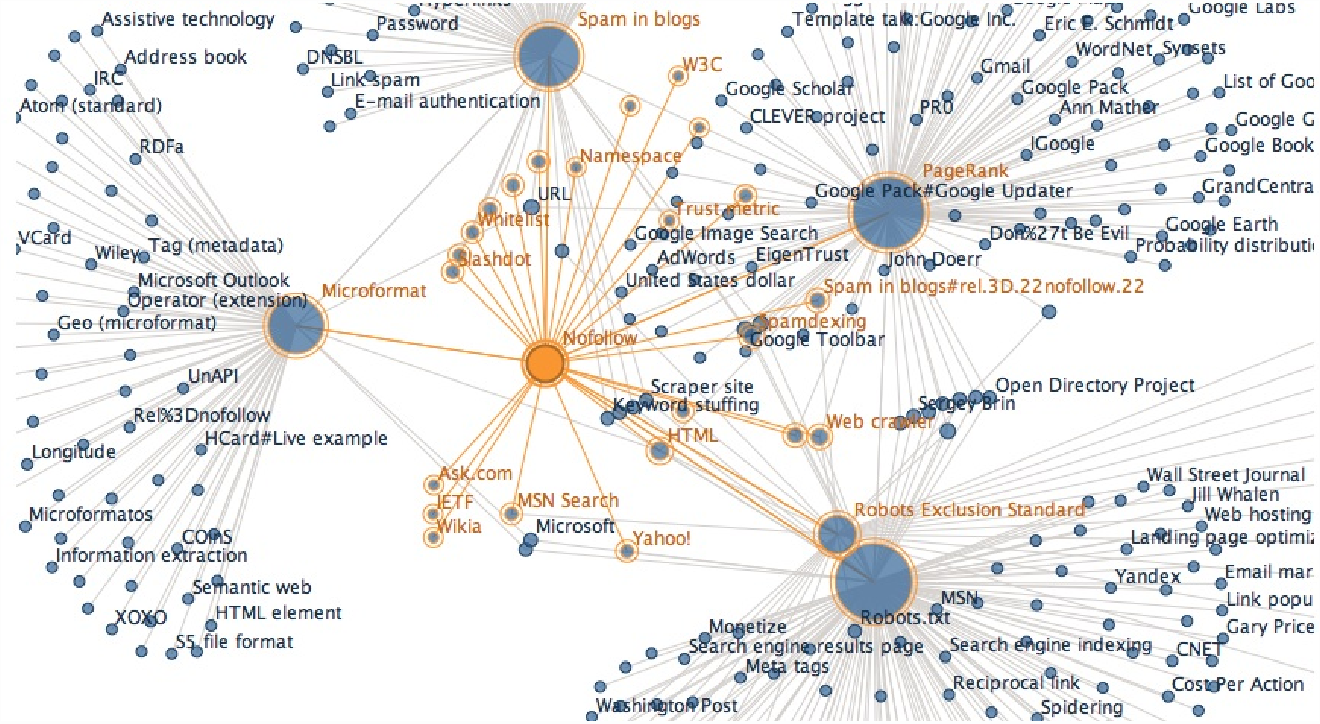
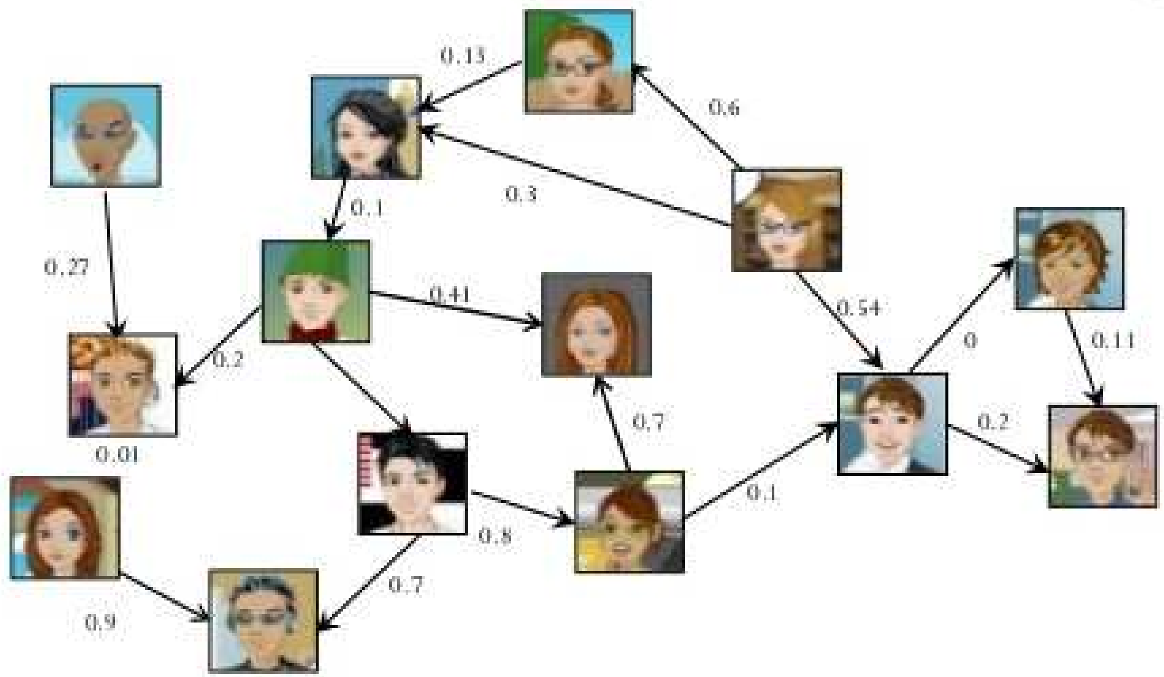


Figure from https://wiki.digitalmethods.net/Dmi/WikipediaAnalysis

Example: social network structure



- **Nodes**: Individuals in the network
- **Edges**: Links/relationships between individuals
- **Edge weight on (i, j)**: Influence weight $w_{i,j}$

Source: Lu and Lakshmanan ICDM 2012
https://www.slideshare.net/WeiLu12/profit-maximization-over-social-networks

Text data

- raw text is a string, i.e., dependency-oriented
- often represented as a **bag-of-words** or **document-term matrix** (nondependency-oriented)
- which can be presented in vector space (as multidimensional data)
  - how often terms occur in document? ⇒ numerical features for term frequencies
  - ⇒ often transformed to tf-idf values (contains weighting + log scaling)

More on the text mining lecture!

Example: tf-idf presentation of sentences

d0: Simple example with cats and mouse
d1: Another simple example with dogs and cats
d2: Another simple example with mouse and cheese

	and	another	cats	cheese	dogs	example	mouse	simple	with
0	1	0	1	0	0	1	1	1	1
1	1	1	1	0	1	1	0	1	1
2	1	1	0	1	0	1	1	1	1

	and	another	cats	cheese	dogs	example	mouse	simple	with
0	0.0	0.000000	0.067578	0.000000	0.000000	0.0	0.067578	0.0	0.0
1	0.0	0.057924	0.057924	0.000000	0.156945	0.0	0.000000	0.0	0.0
2	0.0	0.057924	0.000000	0.156945	0.000000	0.0	0.057924	0.0	0.0

Example from `https://medium.com/@MSalnikov/text-clustering-with-k-means-and-tf-idf-f099bcf95183`

Data preprocessing: main tasks

1. Data cleaning: handling errors and missing values
2. Feature extraction: creating new features by combining and transforming existing ones
   - a **crucial step!** $\Rightarrow$ what patterns you can find
   - application specific $\Rightarrow$ understanding the domain
3. Data reduction
   - sampling
   - feature selection
   - dimension reduction by transformations

1. Data cleaning

Goal: detect & eliminate errors, missing values, duplicates, noise, sometimes outliers

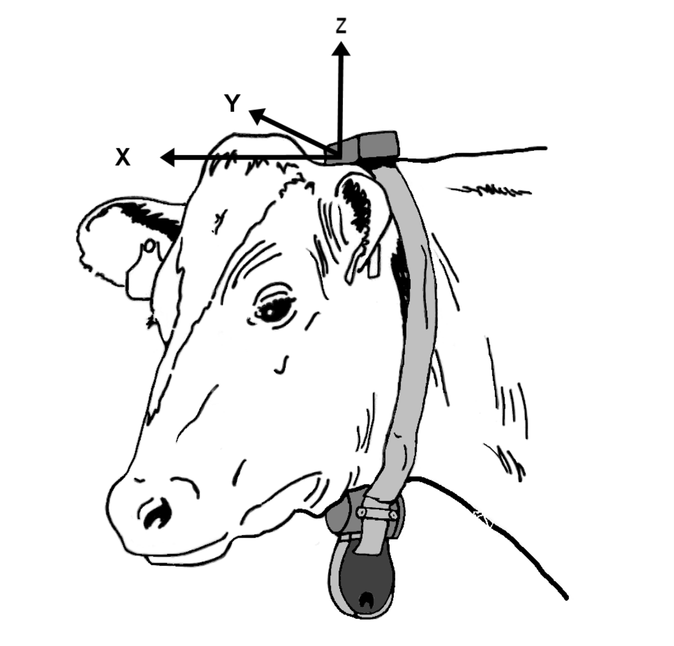
- but outliers may also reveal some interesting event!

Sources:

- automatic measuring devices may stop reading or transmit duplicates (e.g., HW failures or battery exhaustion)
- users may not want to specify (correct) information for privacy reasons
- manually entered data contains very often errors!
- automatically produced text (from scanned documents or speech) prone to errors

Real world example

Task: predict cows' activities (walking, standing, lying, ...)
Data: sequences of accelerometer measurements for time intervals when an animal performs an activity (class).

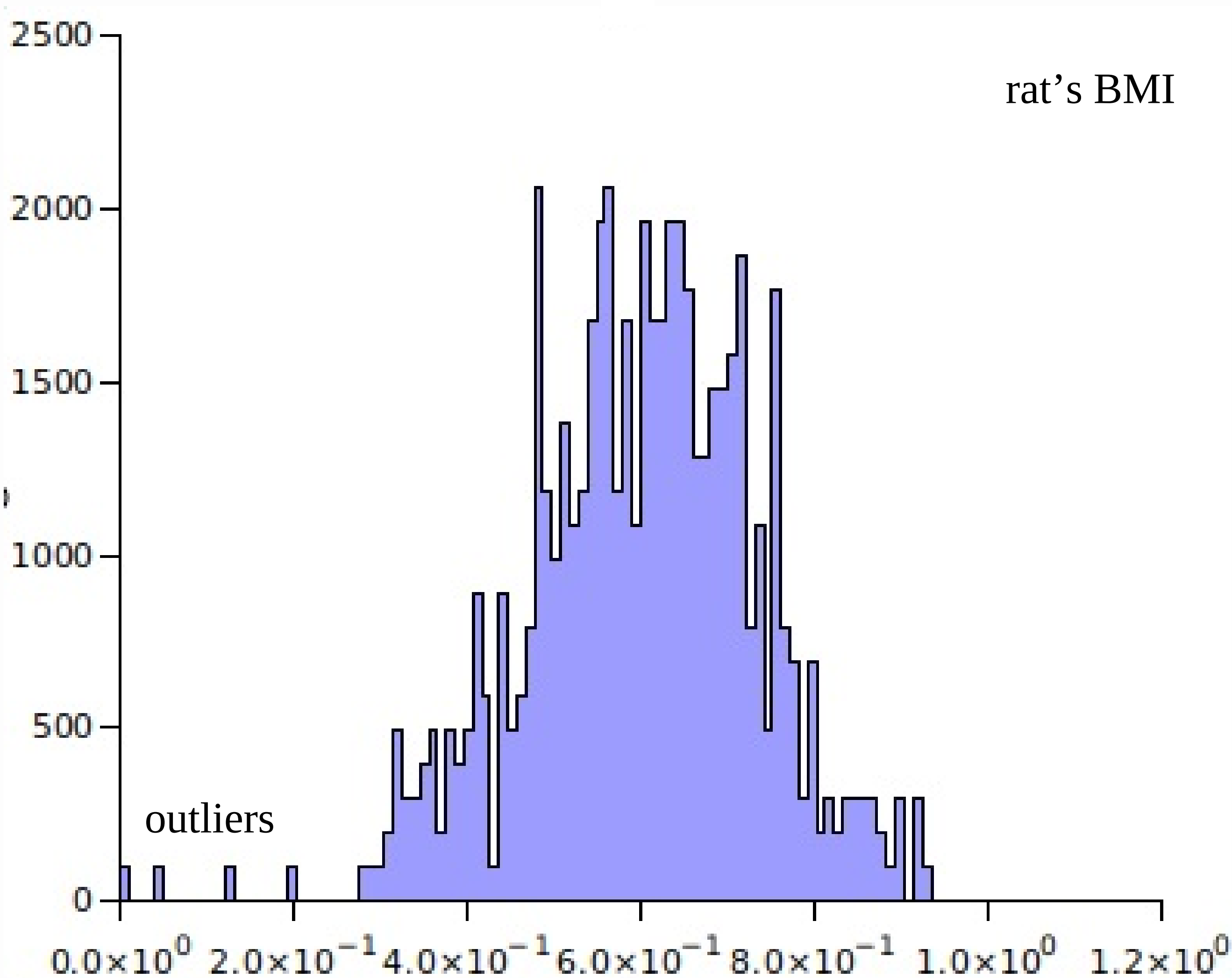


- faulty devices
- lack of calibration
- transmission breaks
- noise
- individual fluctuations
- **errors in human annotation**

Errors and inconsistencies: strategies

- check inconsistencies between different data sources
  - e.g., name spelling
- use domain knowledge
  - known ranges of values (age cannot be 800 yrs)
  - known relationships (if country='USA', city$\neq$'Sanghai')
- check outliers and extreme values (error candidates)
  - not errors, if they have a reasonable explanation
- data smoothing reduces noise and random fluctuations
  - e.g., scaling, discretization, dimension reduction
- use robust methods in the modelling phase

Example: outliers may reveal errors



Missing values: strategies

If possible, **replace with correct values**. Otherwise,

- if a feature has many missing values, **prune the feature**
- if a record has many missing value, **prune the record**
- **impute** missing values
  - mean or median of the feature (among all or similar records/nearest neighbours)
  - predict the missing value using other features (e.g., random forests imputation)
  - **Warning!** Imputation may have a strong effect the results!
- use a **modelling technique that allows missing values** (just replace with special values like "NA")

2. Feature extraction methods

- scaling and normalization: numerical → numerical
- discretization: numerical → categorical
- binarization: categorical → binary (0/1)
- creating similarity graphs: any type → graph
- transformations for dimension reduction: create new less redundant features and keep the best ones
    - both feature extraction + data reduction

Scaling and normalization

Problem: Features with large magnitudes often dominate

$\Rightarrow$ transform to the same scale or standardize distributions

- **min-max scaling**:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)} \quad \text{(new range } [0, 1])$$

- **mean normalization**:

$$y = \frac{x - mean(x)}{\max(x) - \min(x)} \quad \text{(new range } [-1, 1], mean(y) = 0)$$

- **Beware!** outliers can affect a lot!

Standardization or z-score normalization

If the distribution is normal:

$$z = \frac{x - mean(x)}{stdev(x)}$$

$mean(z) = 0$

$stdev(z) = 1$

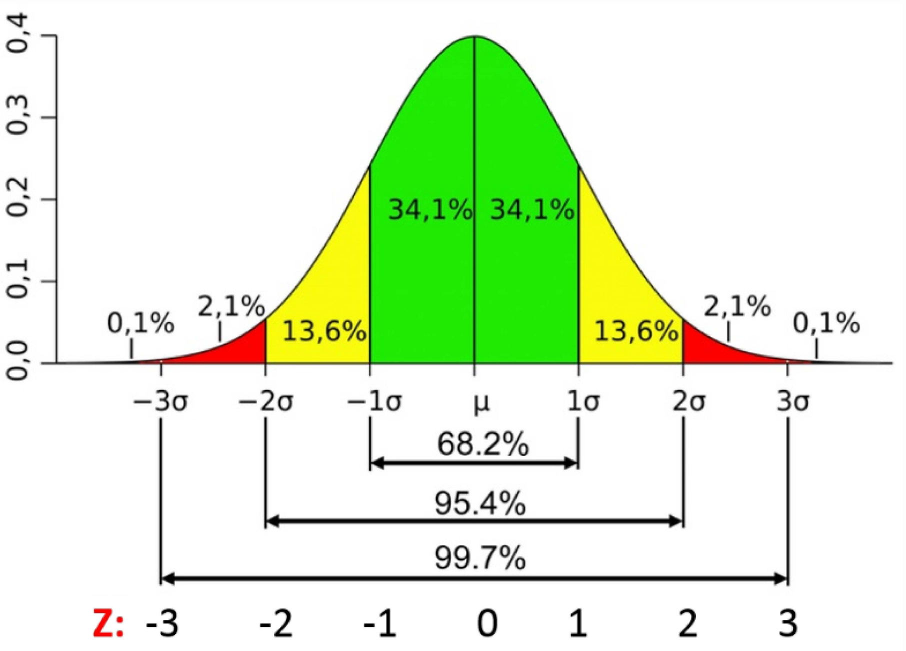


image source: https://sphweb.bumc.bu.edu/otlt/MPH-Modules/PH717-QuantCore/PH717-Module6-RandomError/PH717-Module6-RandomError5.html

Discretization: numerical → categorical

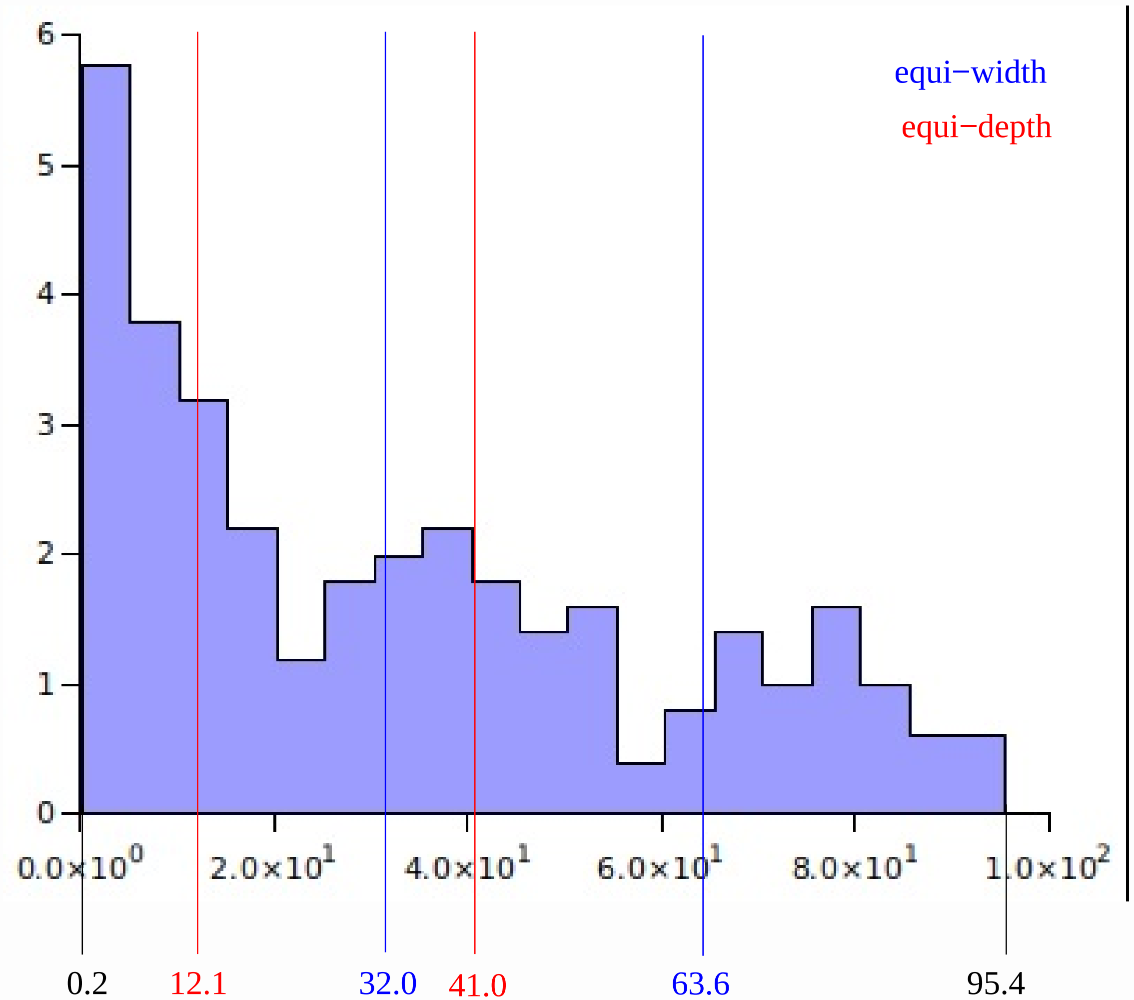
- divide the numerical range into intervals (bins) + give labels to bins
- temperature could be discretized as $T < 0°$C cold, $0 \leq T < 15°$C cool, $T \geq 15°$C warm
- **binarization**: a special case when the new variable is binary (true/false or 1/0)
- e.g., frost=1, if $T < 0$ and frost=0 otherwise
- Note! Also categorical variables can be binarized
  - eye-colour={blue, brown, green, grey} ⇒ blue-eyed=1, if eye-colour=blue, and 0 otherwise

Some discretization methods

- Equi-width discretization
  - equally wide bins
  - good if uniform distribution
- Equi-depth (equal frequency)
  - each bin has an equal number of records
- Many supervised methods if class labels available
- Visual/manual: often best results, but can be worksome

Example: internet users/100 people in countries

Equi-width or equi-depth wouldn't present natural groups



Discretization: benefits and limitations

+ good way to handle mixed data
+ removes noise and individual variation

$\Rightarrow$ it is often worth of analyzing a discretized version of purely numerical data

  + less noise, clearer patterns
  + more efficient algorithms
  + discrete patterns may help to choose the right modelling method also for numerical data

- loses some information
- optimal discretization difficult! (optimal discretization of one variable may depend on other variables)

Useful type transport: any type → similarity graph

- idea: present **pairwise similarities** among closest neighbours by a neighbourhood/similarity graph
- suitable for **any data type** if the distance/similarity function can be defined
- for any application based on the notion of similarity/distances
  - e.g., clustering, recommendations based on similarity
- enables use of numerous network algorithms
- Beware: can be time consuming for large data! (brute force $\mathcal{O}(n^2)$, n=number of objects)

Constructing nearest neighbour graph (idea)

Given objects $O_1, \ldots, O_n$, a distance measure d and a user-defined parameter ϵ or K.

1. create a node for each O_i
2. create an edge between a pair near/similar objects:
   i) if $d(O_i, O_j) \leq \epsilon \Rightarrow$ undirected edge $O_i - O_j$ **or**
   ii) if O_j is among K nearest neighbours of $O_i \Rightarrow$ directed edge $O_i \rightarrow O_j$ (direction can be ignored)
3. give weights to edges reflecting similarity, e.g.,

$$w_{ij} = e^{-d(O_i,O_j)^2/t^2} \quad \text{(heat kernel, } t \text{ user-defined)}$$

3. Data reduction: approaches

1. sampling (select a subset of records)
2. feature selection (select a subset of features)
   - application specific!
   - **filtering** methods: prune features before modelling
   - **wrapper** methods: use modelling (e.g., clustering) to evaluate goodness of feature sets
   - **hybrid** methods: candidates by filtering + evaluation by modelling
3. dimension reduction
   - by axis rotation (PCA, SVD)
   - with type transformation

Main messages

- careful with data types
- careful with preprocessing (data often dirty!)
- feature extraction has a strong effect

Reading for lecture 1:

Book Ch 1 and Ch 2 except 2.4.3–2.4.4