DM & ML
Input: Instances, attributes

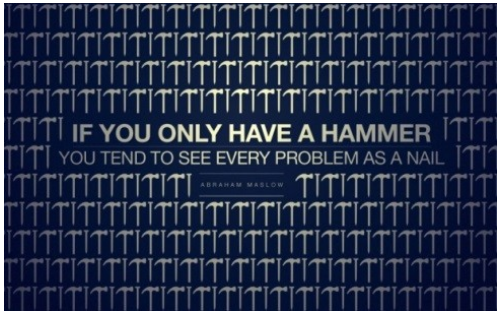
Gergely Lukács
Pázmány Péter Catholic University
Faculty of Information Technology and Bionics
Budapest, Hungary
lukacs@itk.ppke.hu

Contents

- Instances
- Attributes
  - Nominal
  - Ordinal
  - Interval
  - Ratio
- ARFF file format

Instances, attributes

- Instance: specific type of example
  - Thing to be classified, associated, or clustered
  - Individual, independent example
  - Characterized by a predetermined set of attributes
- Input to learning scheme: set of instances/dataset
  - Represented as a single table/"relation"/flat file
  - One line (vector) for each instance



Independent? Limitations

- Independent examples
  - In practical situation? Limited data, financial/time/… constraints on gathering data
  - E.g. music genre recognition 3 sec pieces of same track?; medical diagnosis: patients of same doctor/hospital?
- Flat table: most common, but pretty restricted form!
  - analog signal (quantisation, sampling, windowing, time + spectral features,.. -> flat table)
  - time-series (differences, derivation,… -> flat table)
  - multiple-instance learning
  - …

Attribute types

- Each instance is described by a fixed predefined set of features, its “attributes”
- Possible attribute types (“levels of measurement”):
  - *Nominal, ordinal, interval* and *ratio*

Nominal quantities

- Values are distinct symbols, „categories”
  - Values themselves serve only as labels or names
  - *Nominal* comes from the Latin word for name
  - Special type: binary, only two categories
- Examples:
  - marital status (single, married, divorced),
  - gender (male/female)
- No relation is implied among nominal values (no ordering or distance measure)
- **Only equality tests** can be performed

Ordinal quantities

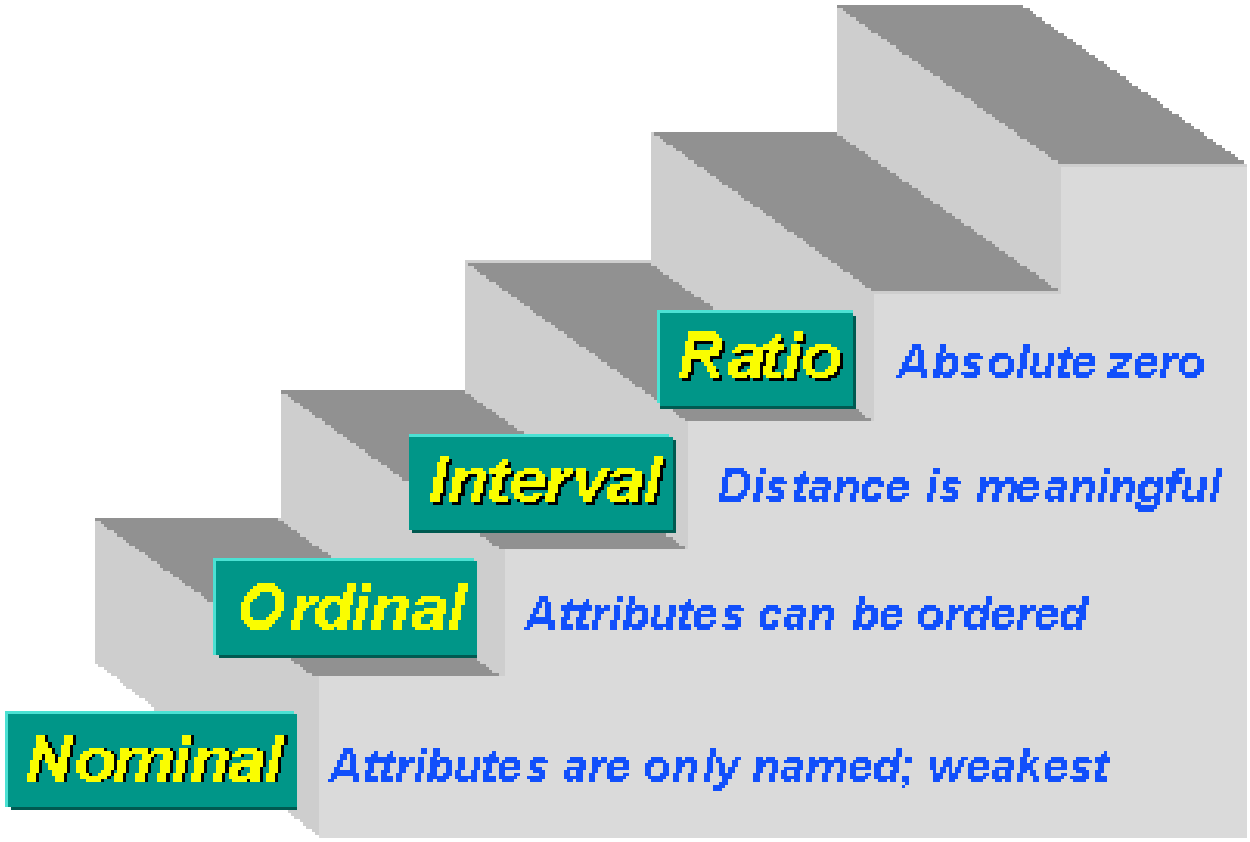
- Impose **order on values**
- **Comparisons** possible (<,>,=,…)
- But: no distance between values defined
- Example:
  - Satisfaction: very unsatisfied < satisfied < very satisfied
  - temperature: cool< mild < hot
- Note: addition and subtraction don’t make sense
- Example rule: temperature < hot c play = yes

Interval quantities

- Interval quantities are not only ordered but **measured in fixed and equal units**
- Examples
  - attribute “temperature” expressed in degrees Fahrenheit
    - the difference between 70 and 75 degrees is the same as the difference between 75 and 80 degrees.
    - You cannot say that 80 degrees is twice as hot as 40 degrees because the zero point on an interval scale is arbitrary
  - attribute “year”
- **Difference** of two values makes sense
- Product doesn’t make sense
- Sum:
  - difference of two values + third value: OK
  - two values: not defined

Ratio quantities

- Ratio quantities are ones for which the measurement scheme defines a **zero point**
- Examples:
  - Distance
  - Age in years
- Ratio quantities are treated as real numbers
  - **All mathematical operations** are allowed


Ratio
Absolute zero
Interval
Distance is meaningful
Ordinal
Attributes can be ordered
Nominal
Attributes are only named; weakest

=, ≠, ‹, › , -, +,*, /

=, ≠, ‹, › , -, ((+))

=, ≠, ‹, ›

=, ≠

Attribute type conversions

- Many algorithms (and tools) accommodate just two levels of measurement (or even directly just one!)
  - nominal (special case: boolean)
  - numeric
- Conversions:
  - nominal->n-1 binary-nominal attributes:
    separate binary attributes for each nominal value
    - Python scikit learn: one-hot encoder
  - nominal -> numeric attribute:
    integers are assigned to the nominal values
    - Python sklearn ordinal encoder
    - **False assumptions**: order, difference on numeric attribute!
  - ordinal -> *n-1* Boolean attributes

Temperature
Cold
Medium
Hot

→

Temperature > cold	Temperature > medium
False	False
True	False
True	True

  - ordinal -> nominal attribute: information on **the order of values lost**!

Inaccurate values

- Data has not been collected for mining it
- => errors and omissions that don't affect original purpose of data (e.g. age of customer)

- Typographical errors in nominal attributes
  ⇒ values need to be checked for consistency
- Typographical and measurement errors in numeric attributes
  ⇒ **outliers** need to be identified
- Other problems: duplicates, stale data

Missing values

- Types: unknown, unrecorded, irrelevant
- Reasons: malfunctioning equipment, changes in experimental design, collation of different datasets, measurement not possible, …
- In data sources: frequently indicated by out-of-range entries (db: NULL)
- Two types:
  - Missing by chance, non-systematic missing
  - Missing value has ***significance*** in itself
    - e.g. classification male/female, attribute age – males are more precise about their age
    - most algorithms assume non-systematic missing ⇒ “missing” may need to be coded as additional value

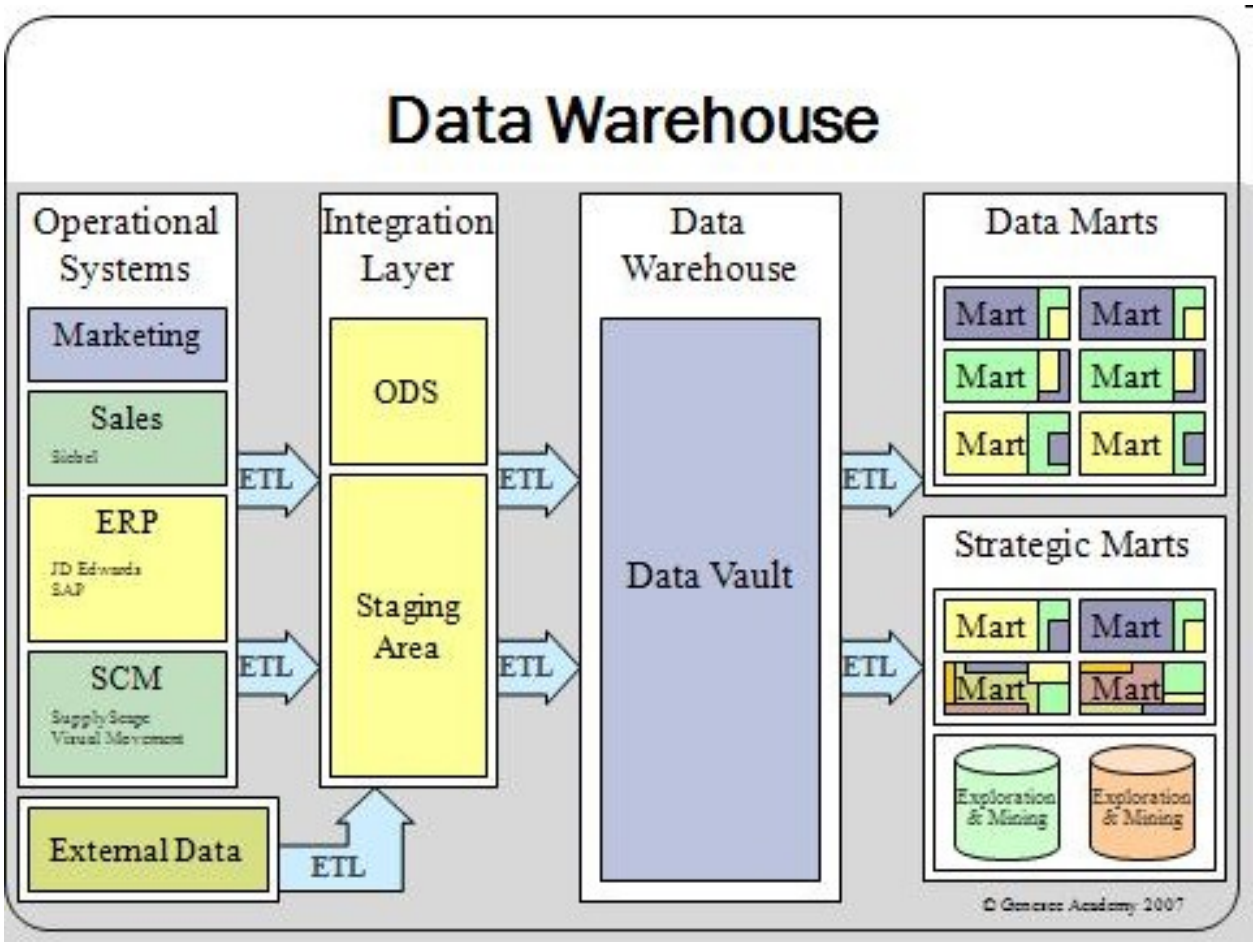
Denormalisation

- Process of flattening called ***denormalization***
  - Several tables („relations” in database terms) are joined together to make one
- Denormalization may produce spurious regularities that reflect structure of database
  - Example: “supplier” predicts “supplier address”

Data cleansing, data integration

- Problem: **different data sources** (e.g. sales department, customer billing department, …)
  - Differences: styles of record keeping, conventions, time periods, data aggregation, primary keys, errors
  - Data must be assembled, integrated, cleaned up
  - “Data warehouse”: consistent point of access
- **External data** may be required (“overlay data”)
- Critical
  - Errors, cleansing
  - type and level of data **aggregation**

(Data Warehouse)



Getting to know the data

- Simple **visualization !!!** tools are very useful for identifying problems
  - Nominal attributes: histograms (Distribution consistent with background knowledge?)
  - Numeric attributes: diagrams (Any obvious outliers?)
- 2-D and 3-D visualizations show dependencies

- (R, Python, Tableau, PowerBI, Weka)

Getting to know the data 2

	Central location	Dispersion
Nominal	Mode	Information only
Ordinal	Median	Percentages
Interval	Arithmetic Mean	Standard or Average Deviation
Ratio	Geometric or Harmonic Mean	Percent Variation

Getting to know the data 3

- Too much data to inspect? Take a **sample**!
  - (~ *"The single most important factor in the quality of an individual's software development is the length of the compile/debug cycle„ – similar in data analysis)*

- **Domain experts** need to be consulted!!
  - In combination with data inspection

Summary

- Standard form of data for data mining
- Instance
  - independence
- Types of attributes/Levels of measurement
  - name, description, operations
  - conversions
    - lost information, false assumption
- Missing values
  - Missing: meaning or just by chance?
- Getting to know your data