

Data Mining Process

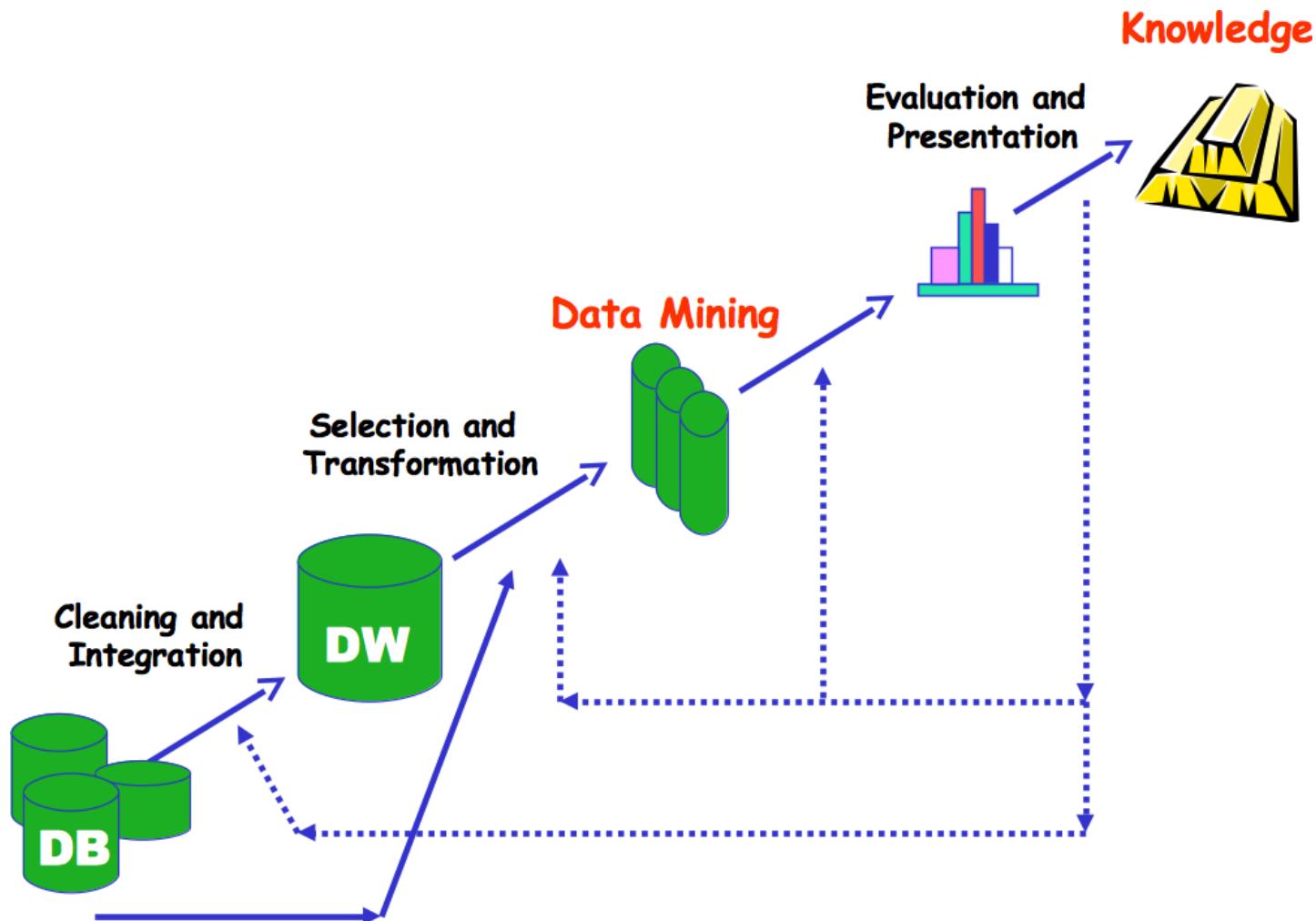
Carlos Soares

(partly using materials kindly
provided by José Luís Borges)

reminder: data mining?

- (or Knowledge Discovery in Databases)
- Is the non-trivial process of identifying
 - implicit (by contrast to explicit)
 - valid (patterns should be valid on new data)
 - novel (novelty can be measured by comparing to expected values)
 - potentially useful (should lead to useful actions)
 - understandable (to humans)
- patterns in data
- Data Mining
 - is a step in the KDD process
 - (arguable, but who cares anyway!...)

the KDD process



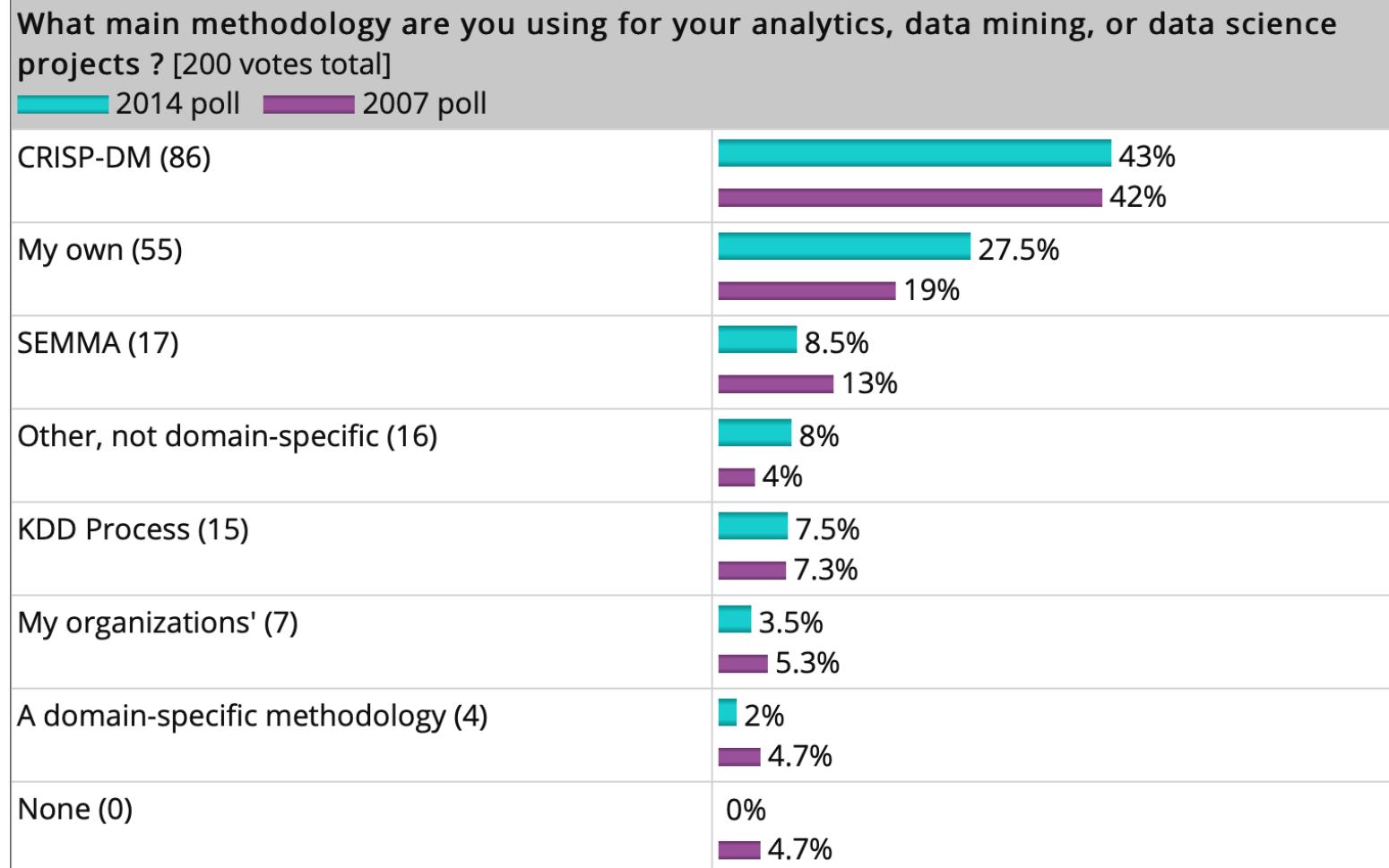
wanted: DM methodology

- Framework for recording experience
 - Allows projects to be replicated
- Aid to project planning and management
- “Comfort factor” for new adopters
 - Demonstrates maturity of Data Mining
 - Reduces dependency on “stars”
- Encourage best practices and help to obtain better results

plan

- methodologies
 - CRISP-DM
 - SCRUM-DM
- ... and beyond
 - MLOps

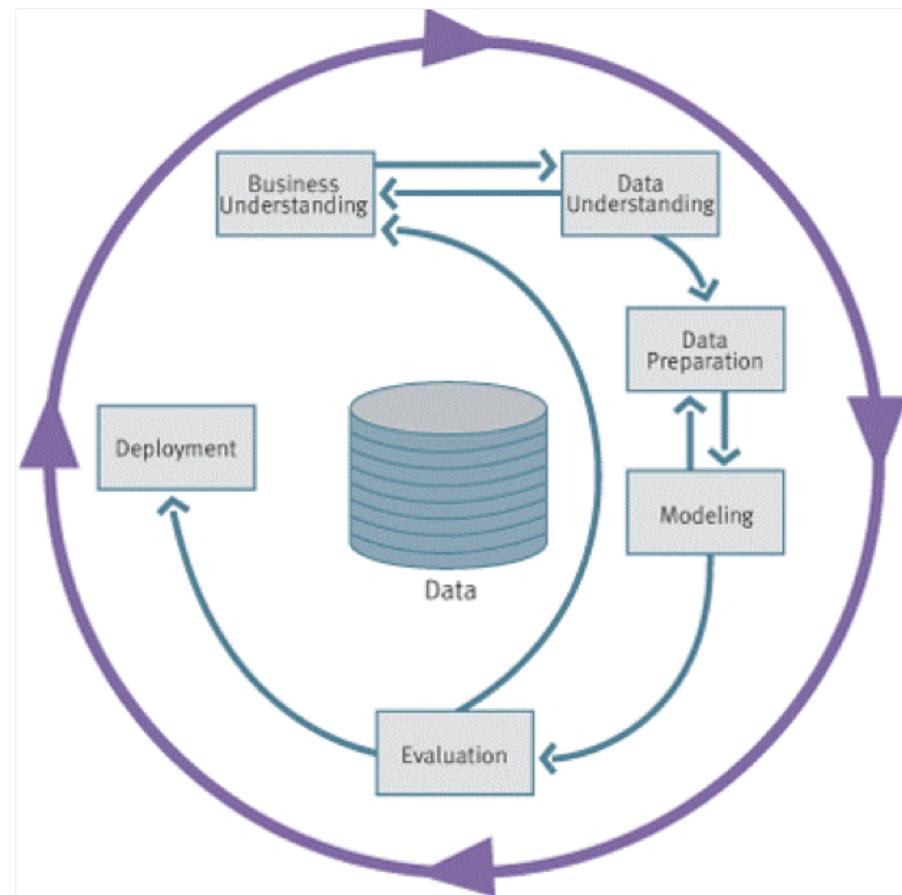
why CRISP-DM?



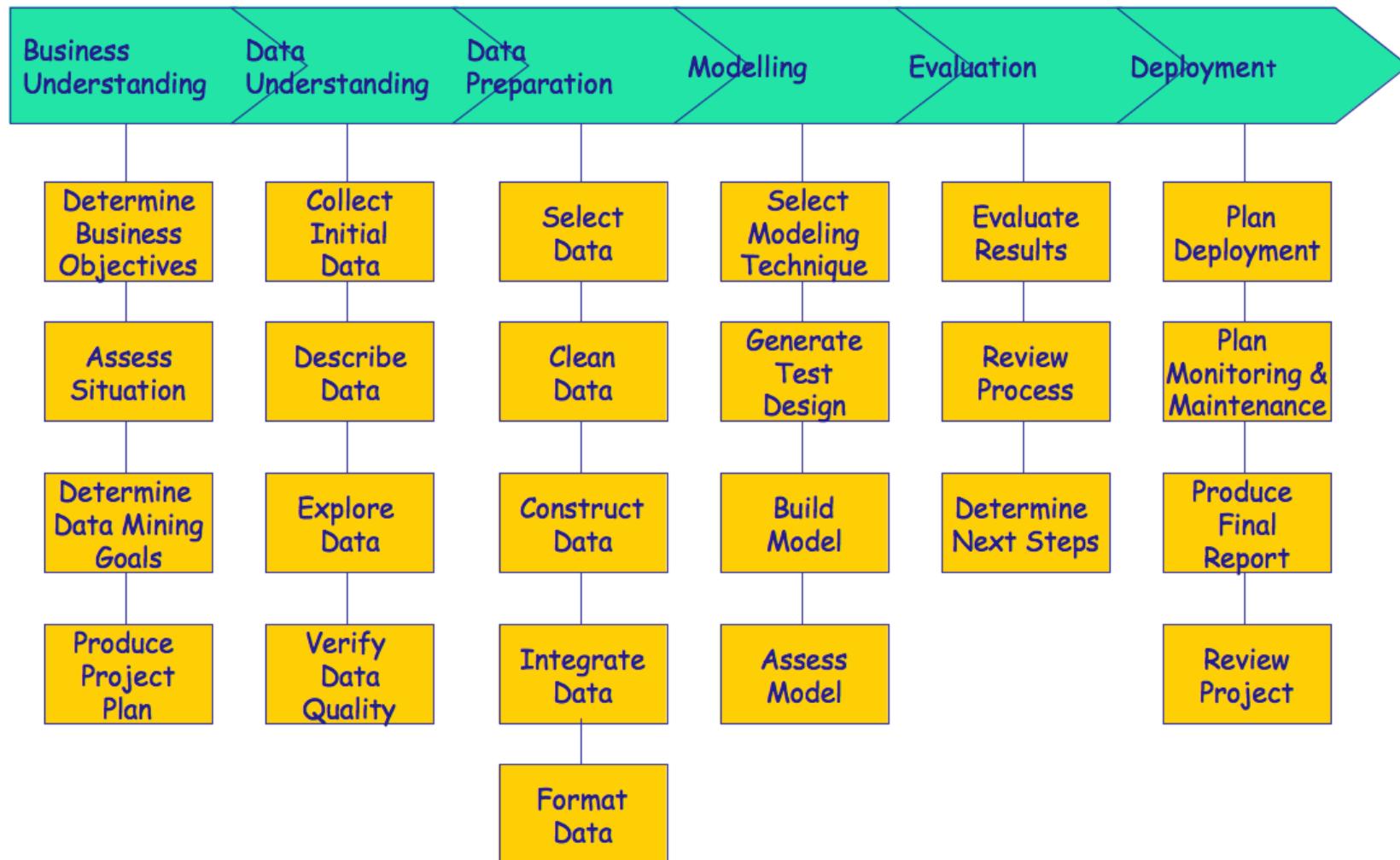
<https://www.kdnuggets.com/polls/2014/analytics-data-mining-data-science-methodology.html>

CRISP-DM: overview

- cross-Industry Standard Process for Data Mining (CRISP-DM)
- European Community funded effort aiming to
 - cheaper, faster, and more reliable data mining
 - widespread adoption
 - reduce skills required for data mining
 - capture experience for reuse
- characteristics
 - non-proprietary
 - application/industry neutral
 - tool neutral
 - focus on business issues
 - as well as technical analysis
 - framework for guidance
 - experience based
 - templates for analysis



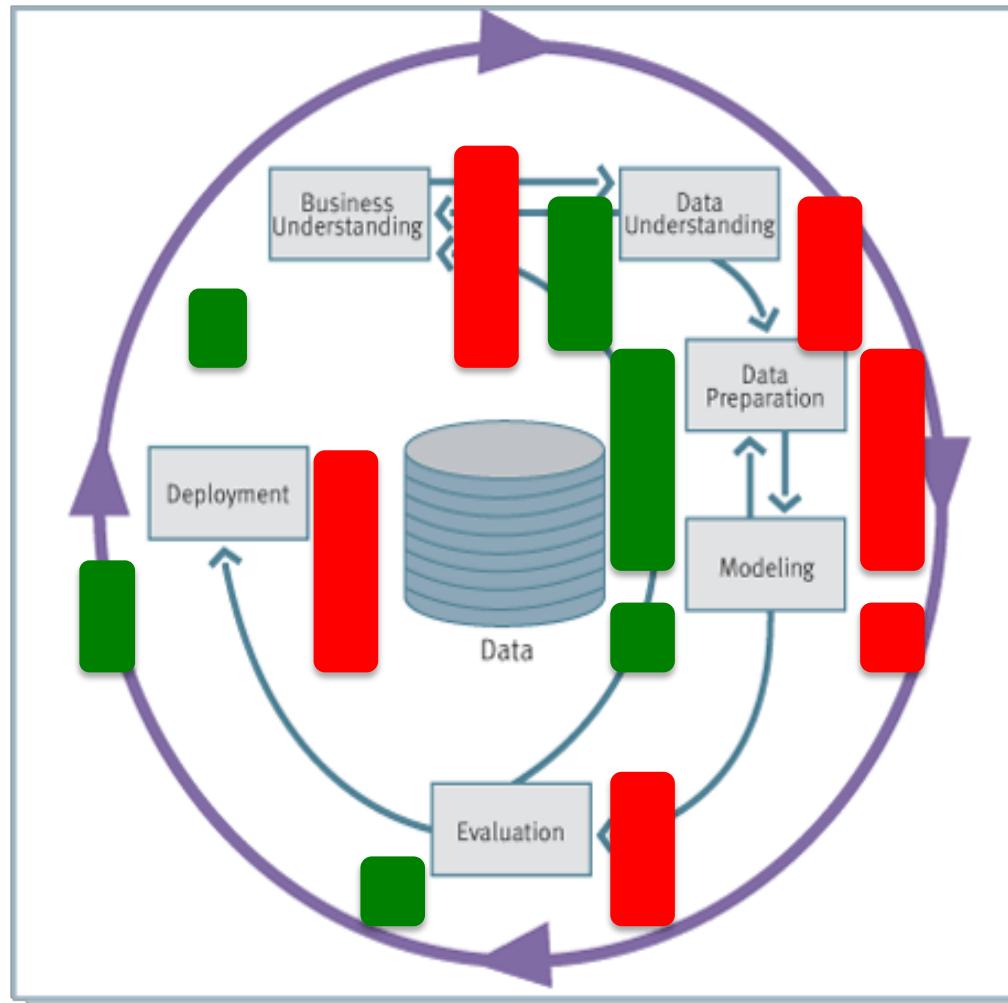
CRISP-DM: phases and tasks



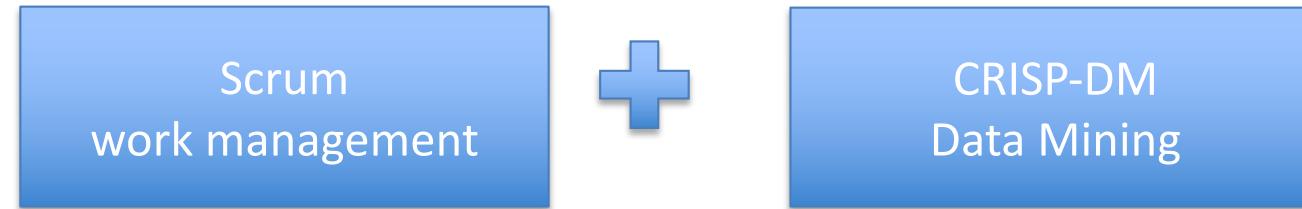
other methodologies

- SEMMA
 - <http://www.sas.com/offices/europe/uk/technologies/analytics/datamining/miner/semma.html>
 - SAS Enterprise Miner
- Others
 - specific
 - <http://datalligence.blogspot.com/2008/12/data-mining-methodologies.html>
- Essentially equivalent

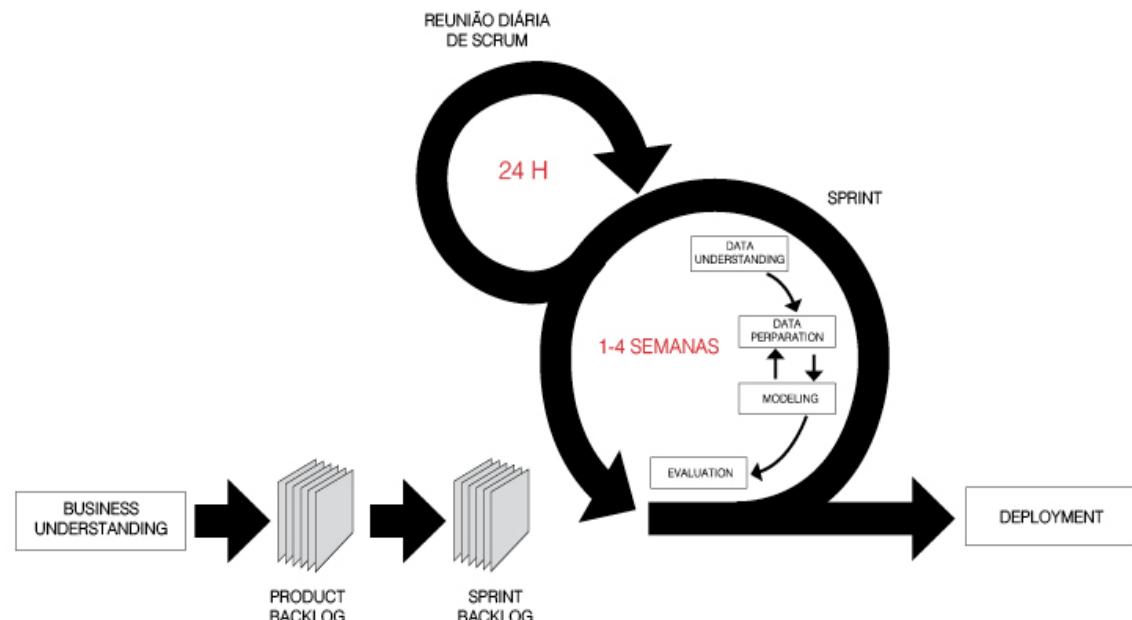
effort/impact on success



SCRUM-DM: an agile DM methodology

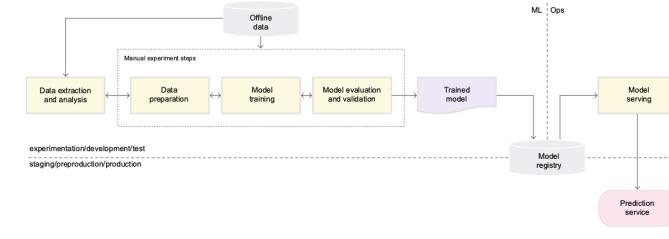


- 3 phases
 - Business Understanding
 - Sprint
 - Deployment
- 6 concepts
 - Product Owner
 - Scrum Master
 - Development team
 - Data Mining Story
 - Product Backlog
 - Sprint Backlog



AI/ML is not (only) software engineering

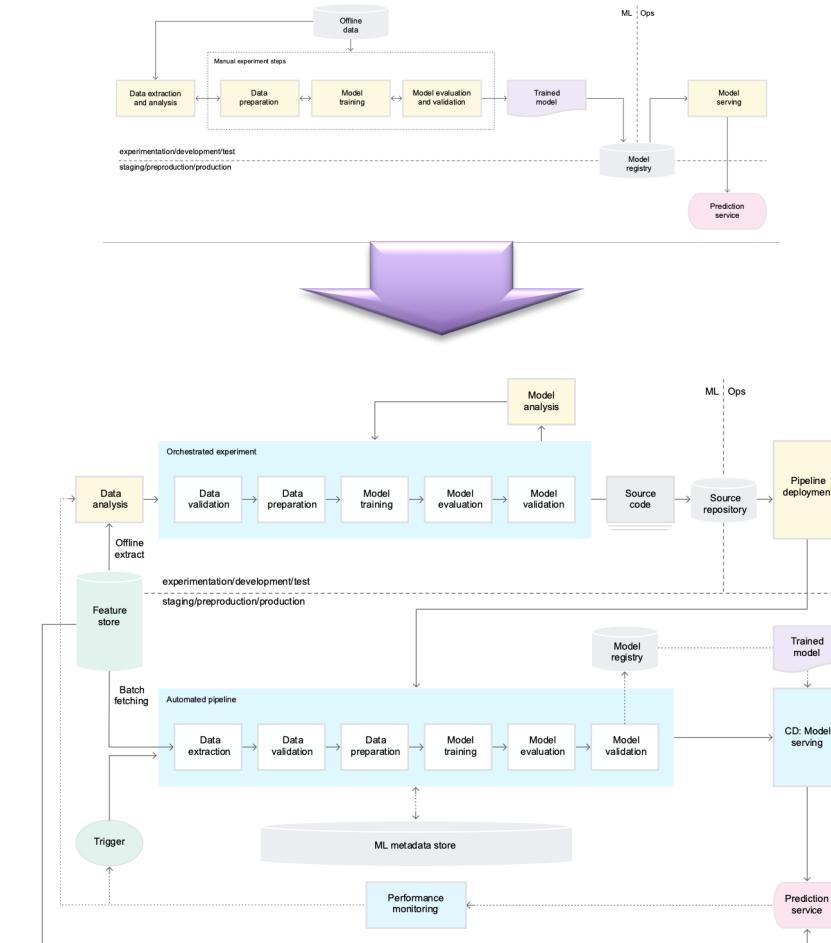
- versioning
 - data
 - model
- automation
 - development
 - testing
- collaborative model development
- deployment
 - monitoring
 - maintenance



source: <https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>

the time for AI/MLOps

- methodologies
 - Microsoft's Team Data Science Process
 - <https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview>
 - Google's Practitioners guide to MLOps: A framework for continuous delivery and automation of machine learning
 - https://services.google.com/fh/files/misc/practitioners_guide_to_mlops_whitepaper.pdf
- tools
 - MLFlow
 - <https://mlflow.org/>
 - Kubeflow
 - <https://www.kubeflow.org/>
 - Data Version Control & Studio
 - <https://studio.iterative.ai/>



source: <https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>

Don't forget

- Curb your enthusiasm...
- A data mining project should always start with an analysis of the data with traditional query tools
 - 80% of the interesting information can be extracted using SQL
 - how many transactions per month include item number 15?
 - show me all the items purchased by Sandy Smith.
 - 20% of hidden information requires more advanced techniques
 - which items are frequently purchased together by my customers?
 - how should I classify my customers in order to decide whether future loan applicants will be given a loan or not?
- Developing and deploying are entirely different beasts!

Data Preparation

Rita P. Ribeiro

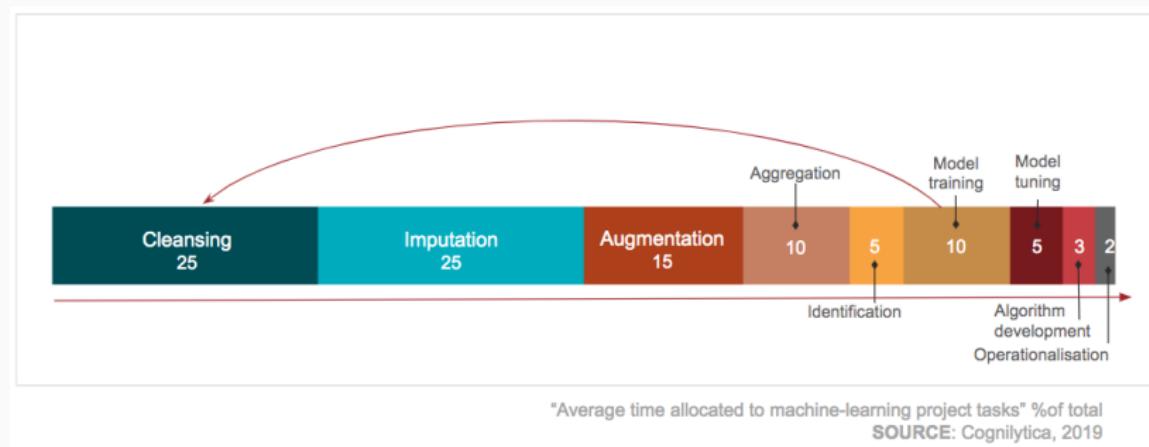
Machine Learning - 2021/2022



DEPARTAMENTO DE CIÉNCIA DE COMPUTADORES
FACULDADE DE CIÉNCIAS DA UNIVERSIDADE DO PORTO

Data Preparation

- The majority of time taken by any data mining project is spent in data preparation
 - e.g. importing, manipulating, cleaning, transforming, augmenting



Data

What is Data?

Collection of data objects (cases) and their attributes (features)

- Attribute: a property or characteristic of an object
 - date, country, temperature, precipitation
- Object: described by a collection of attributes
- It can be structured (e.g. data table) or non-structured (e.g. text)
- It can have non-dependency or dependency between objects (e.g. time, space)

What is Data? (cont.)

Types of data sets

- Nondependency-oriented data
 - the cases do not have any dependencies between them
 - examples: simple data tables, text
- Dependency-oriented data
 - implicit or explicit relationships between cases
 - examples: time series, discrete sequences, spatialtemporal data, network and graph data.

Data: Types of Attributes

- Categorical / Qualitative Attributes
 - Nominal: there is no relationship between the values
 - name, gender, patient id
 - Ordinal: there is an order between the values, but no mathematical operation can be performed on them
 - size $\in \{small, medium, large\}$
- Numeric / Quantitative Attributes
 - Discrete: finite or countably infinite set of values for which differences are meaningful
 - temperatures in Celsius, calendar dates, event duration in minutes
 - Continuous: infinite set of values that represent the absolute numbers
 - number of visits to the hospital, distance, income

Data: Important Characteristics

- Dimensionality (i.e. number of attributes)
 - high dimensional data brings several challenges
- Sparsity
 - presence attributes
- Resolution
 - patterns depend on the scale
- Size
 - type of analysis may depend on size of data

Data Preparation

- Typically, data analysis tasks use source data sets stored in tabular format.
 - datasets are bi-dimensional structures (e.g. table)
- How can we **import data** from different sources and / or formats?
- How can we easily **manipulate the data**?
- How can we **transform the data**?

Data Wrangling

- Process of transforming and mapping data from one “raw” data form into another format appropriate for analytics.
- Main steps
 - discovering
 - structuring
 - cleaning
 - enriching
 - validating
 - publishing
- Goal: attain quality and useful data.

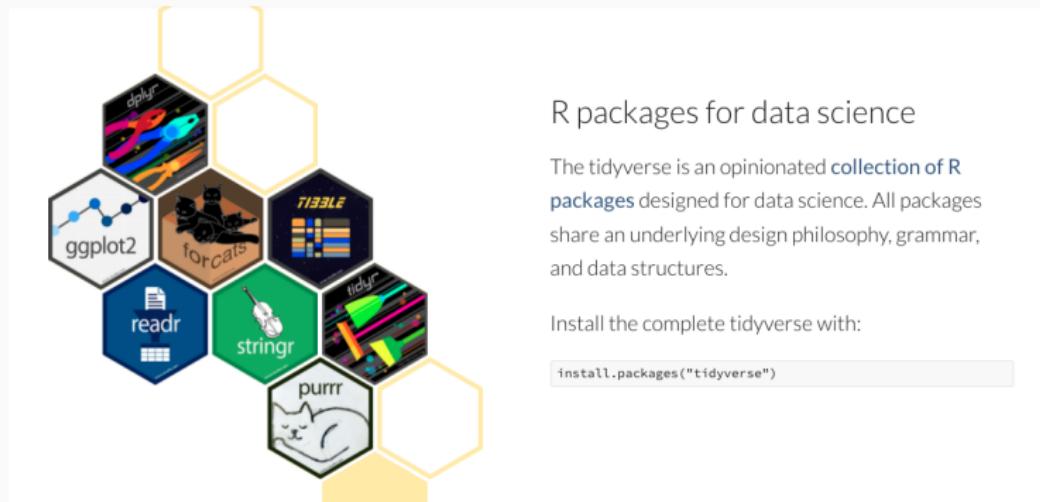
Data Wrangling in R

Data Objects

- Tidy data:
 - every column is variable
 - every row is an observation
 - every cell is a single value
- Data objects: tibbles
 - `int`: integers.
 - `dbl`: doubles, or real numbers.
 - `chr`: character vectors, or strings.
 - `dttm`: date-times (a date + a time).
 - `lgl`: logical, vectors that contain only TRUE or FALSE.
 - `fctr`: factors, i.e. categorical variables with fixed possible values.
 - `date`: dates.

tidyverse - R packages for Data Science

- **tidyverse** - R packages for Data Science



tidyverse - R packages for Data Science (cont.)

- **readr**: provides a fast and friendly way to read rectangular data (e.g. csv)
- **tidyr**: helps you create tidy data
- **stringr**: cohesive set of functions designed to make working with strings as easy as possible.
- **forcats**: provides a suite of tools that solve common problems with factors (categorical variables handled in R)
- **dplyr**: grammar of data manipulation
- **ggplot2**: grammar of graphics

Package `readr`: importing data

```
----- "dummy.csv" -----
ID, Name, Height
23424, Ana, 1.60
11234, Charles, 1.73
77654, Susanne, 1.65
```

```
ds <- read_csv("dummy.csv")
ds

## # A tibble: 3 x 3
##       ID   Name   Height
##   <dbl> <chr>    <dbl>
## 1 23424 Ana      1.6
## 2 11234 Charles  1.73
## 3 77654 Susanne  1.65
```

Package `readr`: importing data (cont.)

```
----- "dummy2.csv" -----
ID; Name; Height
23424; Ana; 1,60
11234; Charles; 1,73
77654; Susanne; 1,65
```

```
ds <- read_delim("dummy2.csv", delim = ";")
ds

## # A tibble: 3 x 3
##       ID   Name  Height
##   <dbl> <chr>   <dbl>
## 1 23424 Ana      160
## 2 11234 Charles    173
## 3 77654 Susanne   165
```

Package `readr`: importing data (cont.)

```
----- "dummy2.csv" -----
ID; Name; Height
23424; Ana; 1,60
11234; Charles; 1,73
77654; Susanne; 1,65
```

```
ds <- read_delim("dummy2.csv", delim = ";", locale = locale(decimal_mark = ","))

## # A tibble: 3 x 3
##       ID   Name   Height
##   <dbl> <chr>    <dbl>
## 1 23424 Ana      1.6
## 2 11234 Charles   1.73
## 3 77654 Susanne  1.65
```

Package `readr`: importing data (cont.)

```
----- "dummy.txt" -----
ID, Name, Height
23424, Ana, ?
11234, Charles, 1.73
77654, Susanne, 1.65
```

```
ds <- read_csv("dummy.txt", na = "?")
ds

## # A tibble: 3 x 3
##       ID   Name   Height
##   <dbl> <chr>    <dbl>
## 1 23424 Ana      NA
## 2 11234 Charles   1.73
## 3 77654 Susanne  1.65
```

Package `readr`: importing data (cont.)

Data import with the tidyverse :: CHEAT SHEET



Read Tabular Data with `readr`

`readr` *`(file, col_names = TRUE, col_types = NULL, col_select = NULL, id = NULL, locale, n_max = Inf, skip = 0, na = c("","NA"), guess_max = min(1000, n_max), show_col_types = TRUE)` See `?readr_delim`

ABC 1,2,3 4,NA

`readr_delim("file.txt", delim = "|")` Read files with any delimiter. If no delimiter is specified, it will automatically guess.

A,B,C 1,2,3 4,NA

`readr_csv("file.csv")` Read a comma delimited file with period decimal marks.

A,B,C 1,2,3 4,NA 5,NA

`readr_semicolon("file.csv")` Read semicolon delimited files with comma decimal marks.

A,B,C 1,2,3 4,NA 5,NA

`readr_tsv("file.tsv")` Read a tab delimited file. Also `read_table`.

`readr_fwf("file.tsv", fef_widths(c(2, 2, 2)))` Read a fixed width file.

USEFUL READ ARGUMENTS

A B C 1 2 3 4 5 NA

No header

`readr_csv("file.csv", col_names = FALSE)`

4 5 6	Skip lines
4 5 6	<code>readr_csv("file.csv", skip = 1)</code>

A B C 1 2 3 4 5 NA

Provide header

`readr_csv("file.csv", col_names = c("x", "y", "z"))`

ABC 1,2,3 4,NA

Read multiple files into a single table

`readr_csv("1.csv", "2.csv", "3.csv", id = "origin_file")`

Save Data with `readr`

`write_r_(x, file, na = "NA", append, col_names, quote, escape, col, num_threads, progress)`

A,B,C 1,2,3 4,NA

`write_delim(x, file, delim = ",")` Write files with any delimiter.

`write_csv(x, file)` Write a comma delimited file.

`write_csv2(x, file)` Write a semicolon delimited file.

`write_tsv(x, file)` Write a tab delimited file.



One of the first steps of a project is to import outside data into R. Data is often stored in tabular formats, like csv files or spreadsheets.

The front page of this sheet shows how to import and save text files into R using `readr`.

The back page shows how to import spreadsheet data from Excel files using `readxl` or Google Sheets using `googlesheets4`.

OTHER TYPES OF DATA

Try one of the following packages to import other types of files:

- `haven` - SPSS, Stata, and SAS files
- `DBI` - databases
- `jsonlite` - json
- `xml2` - XML
- `httr` - Web APIs
- `rvest` - HTML (Web Scraping)
- `readr/read_lines` - text data

Column Specification with `readr`

Column specifications define what data type each column of a file will be imported as. By default, `readr` will generate a column spec when a file is read and output a summary.

`spec(x)` Extract the full column specification for the given imported data frame.

`spec(x)
col1
age = col_logical()
sex = col_character()
earn = col_double()
}
earn is a double (numeric)
sex is a character`

COLUMN TYPES

Each column type has a function and corresponding string abbreviation.

- `col_logical()` - "l"
- `col_integer()` - "i"
- `col_double()` - "d"
- `col_number()` - "n"
- `col_character()` - "c"
- `col_factor(levels, ordered = FALSE)` - "f"
- `col_datetime(format = "%Y-%m-%d %H:%M:%S")` - "T"
- `col_date(format = "%Y-%m-%d")` - "D"
- `col_time(format = "%H:%M:%S")` - "t"
- `col_skip(n = 1)` - "s"
- `col_guess()` - "g"

USEFUL COLUMN ARGUMENTS

Hide col spec message

`readr::readr_csv(file, show_col_types = FALSE)`

Select columns to import

Use names, position, or selection helpers.

`readr::readr_csv(file, col_select = c(lage, earn))`

Guess column types

To guess a column type, `readr::readr_csv(file, guess_max = Inf)` looks at the first 1000 rows of data. Increase with `guess_max`.

DEFINE COLUMN SPECIFICATION

Set a default type

`readr::readr_csv(file, col_type = list(default = col_double()))`

Use column type or string abbreviation

`readr::readr_csv(file, col_type = list(x = col_double(), y = "i", z = "c"))`

Use a single string of abbreviations

`# col_types: skip, guess, integer, logical, character
readr::readr_csv(file, col_type = "iilic")`

Package `readr`: importing data (cont.)

Import Spreadsheets with `readxl`

READ EXCEL FILES

x1	x2	x3	x4	x5
x	2	6		
y	7	NA	8	10

`read_excel(path, sheet = NULL, range = NULL)`
Read a .xls or .xlsx file based on the file extension.
See front page for more read arguments. Also
`read_xls()` and `read_xlsx()`.
`read_excel(path, file_type = "xsl")`

READ SHEETS

s1	s2	s3
1	2	3

`read_excel(path, sheet = NULL)` Specify which sheet to read by name or index.
`read_xls(path, sheet = 1)`
`read_xlsx(path, sheet = "s1")`

`excel_sheets(path)` Get a vector of sheet names.
`excel_sheets("excelfile.xlsx")`

s1	s2	s3
1	2	3

To read multiple sheets:
1. Get a vector of sheet names from `excel_sheets`.
2. Use `map_dfr` to read each sheet.
3. Use `purrr::map_dfr` to read multiple files into one data frame.

```
path <- "your_file_path.xlsx"
path %>% excel_sheets() %>%
  set_names() %>%
  map_dfr(read_excel, path = path)
```

OTHER USEFUL EXCEL PACKAGES

For functions to write data to Excel files, see:
• `openxlsx`
• `writexl`

For working with non-tabular Excel data, see:
• `tidyxl`



with `googlesheets4`

READ SHEETS

x1	x2	x3	x4	x5
x	2	6		
y	7	NA	8	10

`read_sheet(ss, sheet = NULL, range = NULL)`
Read a sheet from a URL, a Sheet ID, or a dribble from the `googledrive` package. See front page for more read arguments. Same as `range_read()`.

SHEETS METADATA

URLs are in the form:
https://docs.google.com/spreadsheets/d/SPREADSHEET_ID/edit#gid=SHEET_ID

`gs4_get(ss)` Get spreadsheet meta data.

`gs4_find(ss)` Get data on all spreadsheet files.

`sheet_properties(ss)` Get a tibble of properties for each worksheet. Also `sheet_names(ss)`.

WRITE SHEETS

TRUE	FALSE	1	2	3	4	5	6	7	8	9	10
TRUE	2	hello	1947-01-06	hello							

- skip
- logical
- date
- guess
- numeric
- list
- text

Use `list` for columns that include multiple data types. See `tidyverse` and `purr` for list-column data.

TRUE	FALSE	1	2	3	4	5	6	7	8	9	10
TRUE	2	hello	1947-01-06	hello							

`sheet_create(ss, ..., sheet = NULL)` Create a new sheet with a vector of names, a data frame, or a (nested) list of data frames.

`sheet_append(ss, data, sheet = 1)` Adds rows to the end of a worksheet.

TRUE	FALSE	1	2	3	4	5	6	7	8	9	10
TRUE	2	hello	1947-01-06	hello							

`sheet_modify(ss, col, cell, value)` Modify a cell's value.

Also use the `range` argument with cell specification functions `cell_limits()`, `cell_rows()`, `cell_cols()`, and `anchored()`.



GODGLESHEETS4 COLUMN SPECIFICATION

Column specifications define what data type each column of a file will be imported as.

Use the `col_types` argument of `read_sheet()` or `range_read()` to set the column specification.

Guess column types

To guess a column type `read_sheet()` or `range_read()` looks at the first 1000 rows of data. Increase with `guess_max`.

`read_sheet(path, guess_max = Inf)`

Set all columns to same type, e.g. `character`
`read_sheet(path, col_types = "text")`

Set each column individually

If col types: skip, guess, integer, logical, character
`read_sheet(ss, col_types = "text")`

COLUMN TYPES

TRUE	FALSE	1	2	3	4	5	6	7	8	9	L
TRUE	2	hello	1947-01-06	hello							

Write a data frame into a new or existing Sheet.

`gs4_create(name, ...)`

`sheet = NULL)`

Create a new sheet with a vector of names, a data frame, or a (nested) list of data frames.

`sheet_modify(ss, col, cell, value)`

Modify a cell's value.

Also use the `range` argument with cell specification functions `cell_limits()`, `cell_rows()`, `cell_cols()`, and `anchored()`.

• skip - " " or "n"

• guess - "??"

• logical - "l"

• integer - "i"

• double - "d"

• numeric - "n"

• date - "D"

• datetime - "I"

• character - "c"

• list-column - "L"

• cell - "C" Returns a list of raw cell data.

Use list for columns that include multiple data types. See `tidyverse` and `purr` for list-column data.

FILE LEVEL OPERATIONS

`googlesheets4` also offers ways to modify other aspects of Sheets (e.g. freeze rows, set column width, manage worksheets). Go to googlesheets4.tidyverse.org to read more.

For whole-file operations (e.g. renaming, sharing, placing within a folder), see the `tidyverse` package `googledrive` at googledrive.tidyverse.org.



RStudio™ is a trademark of RStudio, Inc. © CC BY SA RStudio • info@rstudio.com • 844-448-5222 • rstudio.com • readr.tidyverse.org googlesheets4.tidyverse.org • readr 2.0.0 • readr 1.3.3 • googlesheets4 1.8.8 • Updated: 2021-08

Package `dplyr`: data manipulation

- `dplyr` is very popular package in the R community mostly due to it greatly facilitating the manipulation of data
- Some of its features include:
 - the most basic data manipulation operations are implemented;
 - handles multiple types of data structures (e.g. data frames, databases, ...);
 - but mostly, it's fast!

Package dplyr: data manipulation (cont.)

- **tibble**: a data frame table specifically tailored for the operations of dplyr;

```
library(dplyr)
data(iris)
ir <- as_tibble(iris)
glimpse(ir)

## #> #> Rows: 150
## #> #> Columns: 5
## #> #> $ Sepal.Length <dbl> 5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.6, 5.0, 4.4, 4.9, 5.4, 4
## #> #> $ Sepal.Width  <dbl> 3.5, 3.0, 3.2, 3.1, 3.6, 3.9, 3.4, 3.4, 2.9, 3.1, 3.7, 3
## #> #> $ Petal.Length <dbl> 1.4, 1.4, 1.3, 1.5, 1.4, 1.7, 1.4, 1.5, 1.4, 1.5, 1.5, 1
## #> #> $ Petal.Width  <dbl> 0.2, 0.2, 0.2, 0.2, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.2, 0
## #> #> $ Species      <fct> setosa, setosa, setosa, setosa, setosa, setosa,
```

Package dplyr: basic operations

```
ds2 <- <operation>(ds1, ...)
```

- **filter** : select a subset of rows
- **select** : select a subset of columns
- **arrange** : reorder the rows
- **mutate** : generate new columns
- **summarize** : summarize column values

These basic operations **return a new object** following the intended operation.
They **do not change the object** in the first parameter (the tibble)

Package dplyr: filter

`filter(ds1, cond1, cond2, ...)` returns the rows of the tibble
ds1 satisfying all the conditions cond1, cond2, ...

```
filter(ir, Sepal.Length > 6, Sepal.Width > 3.5)

## # A tibble: 3 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##       <dbl>      <dbl>      <dbl>      <dbl> <fct>
## 1         7.2        3.6        6.1        2.5 virginica
## 2         7.7        3.8        6.7        2.2 virginica
## 3         7.9        3.8        6.4        2     virginica
```

```
filter(ir, Sepal.Length > 7.7 | Sepal.Length < 4.4)
```

```
## # A tibble: 2 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##       <dbl>      <dbl>      <dbl>      <dbl> <fct>
## 1         4.3        3          1.1        0.1 setosa
## 2         7.9        3.8        6.4        2     virginica
```

Package dplyr: select

`select(ds1, col1, col2, ...)` returns the columns `col1`, `col2,...` of the tibble `ds1`

```
select(iris, Sepal.Length, Species)

## # A tibble: 150 x 2
##   Sepal.Length Species
##       <dbl> <fct>
## 1         5.1 setosa
## 2         4.9 setosa
## 3         4.7 setosa
## 4         4.6 setosa
## 5         5.0 setosa
## 6         5.4 setosa
## 7         4.6 setosa
## ...
## # ... other 143 rows
```

Package dplyr: select (cont.)

You can use select in a *negative* way, passing information concerning the columns that the user does not want to select.

```
select(ir, -(Sepal.Length:Petal.Length))

## # A tibble: 150 x 2
##   Petal.Width Species
##       <dbl> <fct>
## 1         0.2 setosa
## 2         0.2 setosa
## 3         0.2 setosa
## 4         0.2 setosa
## 5         0.2 setosa
## 6         0.4 setosa
## 7         0.3 setosa
## ...
## # ... other 143 rows
```

Package dplyr: select (cont.)

If you have a certain number of variables that begin with the same name, you can select them all easily

```
select(ir, starts_with("Sepal"))

## # A tibble: 150 x 2
##       Sepal.Length Sepal.Width
##   <dbl>        <dbl>
## 1     5.1        3.5
## 2     4.9        3
## 3     4.7        3.2
## 4     4.6        3.1
## 5     5           3.6
## 6     5.4        3.9
## 7     4.6        3.4
## ...
## # ... other 143 rows
```

Package dplyr: arrange

`arrange(ds1, col1, col2, ...)` re-arranges the rows of the tibble `ds1` by the user input order (`col1, col2, ...`)

```
arrange(ir, desc(Sepal.Length), Sepal.Width)

## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##       <dbl>      <dbl>      <dbl>      <dbl> <fct>
## 1         7.9        3.8        6.4        2 virginica
## 2         7.7        2.6        6.9        2.3 virginica
## 3         7.7        2.8        6.7        2 virginica
## 4         7.7        3          6.1        2.3 virginica
## 5         7.7        3.8        6.7        2.2 virginica
## 6         7.6        3          6.6        2.1 virginica
## 7         7.4        2.8        6.1        1.9 virginica
## ...
## # ... other 143 rows
```

Package dplyr: mutate

`mutate(ds1, newcol1, newcol2, ...)` adds new columns
(newcol1, newcol2, ...) to the tibble ds1. It does not change the original data.

```
mutate(ir, sr = Sepal.Length/Sepal.Width, pr = Petal.Length/Petal.Width)

## # A tibble: 150 x 7
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species     sr     pr
##       <dbl>      <dbl>      <dbl>      <dbl> <fct>    <dbl>    <dbl>
## 1         5.1        3.5       1.4       0.2 setosa  1.46    7
## 2         4.9        3.0       1.4       0.2 setosa  1.63    7
## 3         4.7        3.2       1.3       0.2 setosa  1.47   6.5
## 4         4.6        3.1       1.5       0.2 setosa  1.48   7.5
## 5         5.0        3.6       1.4       0.2 setosa  1.39    7
## 6         5.4        3.9       1.7       0.4 setosa  1.38   4.25
## 7         4.6        3.4       1.4       0.3 setosa  1.35   4.67
## ...
....
```

Package dplyr: summarize

`summarize(ds1, sumF1, sumF2, ...)` summarizes the rows in the `tibble` data using the user-provided functions `sumF1, sumF2, ...`

```
summarise(ir, avgPL = mean(Petal.Length), varSW = var(Sepal.Width))  
  
## # A tibble: 1 x 2  
##   avgPL  varSW  
##     <dbl> <dbl>  
## 1    3.76  0.190
```

Package dplyr: combining operations

dplyr allows for the combination of basic operations in the same call

```
select(filter(ir, Petal.Width > 2.3), Sepal.Length, Species)

## # A tibble: 6 x 2
##   Sepal.Length Species
##       <dbl> <fct>
## 1         6.3 virginica
## 2         7.2 virginica
## 3         5.8 virginica
## 4         6.3 virginica
## 5         6.7 virginica
## 6         6.7 virginica
```

Package dplyr: combining operations (cont.)

However, composing such functions can become very hard to understand (and code...)

```
arrange(select(filter(mutate(ir, sr = Sepal.Length/Sepal.Width),
      sr > 1.6), Sepal.Length, Species), Species, desc(Sepal.Length))

## # A tibble: 103 x 2
##       Sepal.Length Species
##              <dbl> <fct>
## 1            5     setosa
## 2            4.9    setosa
## 3            4.5    setosa
## 4            7     versicolor
## 5            6.9    versicolor
## 6            6.8    versicolor
## 7            6.7    versicolor
## ...
....
```

Package dplyr: chaining operator

- To provide an easy solution for the combination of `dplyr` operations, one can use the chaining operator (or pipe) `%>%`
- If using this operator, you only need to declare the `tibble` in the first function call
- The chaining operator tells the following operation that the result of the former operation will be the `tibble` to use
- `x %>% f(y)` becomes `f(x, y)`

Package dplyr: chaining operator (cont.)

```
mutate(ir, sr = Sepal.Length/Sepal.Width) %>%
  filter(sr > 1.6) %>%
  select(Sepal.Length, Species) %>%
  arrange(Species, desc(Sepal.Length))

## # A tibble: 103 x 2
##   Sepal.Length Species
##       <dbl> <fct>
## 1         5   setosa
## 2         4.9  setosa
## 3         4.5  setosa
## 4         7   versicolor
## 5         6.9  versicolor
## 6         6.8  versicolor
## 7         6.7  versicolor
## 8         6.7  versicolor
## 9         6.7  versicolor
## 10        6.6  versicolor
## # ... with 93 more rows
```

Package dplyr: group_by

group_by(ds1, crit1, crit2, ...) groups rows of the tibble
ds1 according to user-input criteria crit1, crit2, ...

```
sps <- group_by(iris, Species)
sps

## # A tibble: 150 x 5
## # Groups:   Species [3]
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##       <dbl>      <dbl>      <dbl>      <dbl> <fct>
## 1         5.1      3.5       1.4      0.2  setosa
## 2         4.9      3.0       1.4      0.2  setosa
## 3         4.7      3.2       1.3      0.2  setosa
## 4         4.6      3.1       1.5      0.2  setosa
## 5         5.0      3.6       1.4      0.2  setosa
## 6         5.4      3.9       1.7      0.4  setosa
## ...
## # ... with 144 more rows, and 1 more variable:
## #   Species: fct [3]
```

Package dplyr: group_by (cont.)

You can apply summarize to sub-groups.

```
group_by(ir, Species) %>%
  summarise(mPL = mean(Petal.Length))

## # A tibble: 3 x 2
##   Species      mPL
##   <fct>     <dbl>
## 1 setosa     1.46
## 2 versicolor 4.26
## 3 virginica  5.55
```

Package `tidyr`: basic operations

- `complete`: make implicit missing values explicit
- `drop_na`: make explicit missing values implicit
- `fill`: replace missing values with next/previous value
- `replace_na`: replace missing values with a known value
- `pivot_longer`: “lengthens” data, increasing the number of rows and decreasing the number of columns.
- `pivot_wider`: “widens” data, increasing the number of columns and decreasing the number of rows.

Package `tidyverse`: basic operations (cont.)

```
df <- tibble(x = c(1, 2, NA), y = c("a", NA, "b"))
df
```

```
## # A tibble: 3 x 2
##       x     y
##   <dbl> <chr>
## 1     1     a
## 2     2    <NA>
## 3    NA     b
```

```
df %>%
  drop_na()
```

```
## # A tibble: 1 x 2
##       x     y
##   <dbl> <chr>
## 1     1     a
```

```
df %>%
  drop_na(x)
```

```
## # A tibble: 2 x 2
##       x     y
##   <dbl> <chr>
## 1     1     a
## 2     2    <NA>
```

```
df %>%
  replace_na(list(x = 0,
                  y = "?"))
```

```
## # A tibble: 3 x 2
##       x     y
##   <dbl> <chr>
## 1     1     a
## 2     2     ?
## 3     0     b
```

Package `tidyverse`: basic operations (cont.)

Dataset: Pew religion and income survey with nr of respondees with an income range

```
relig_income

## # A tibble: 18 x 5
##   religion      `<$10k` `'$10-20k` `'$20-30k` `'$30-40k`
##   <chr>        <dbl>     <dbl>     <dbl>     <dbl>
## 1 Agnostic       27        34        60        81
## 2 Atheist         12        27        37        52
## 3 Buddhist        27        21        30        34
## 4 Catholic        418       617       732       670
## 5 Don't know/refused  15        14        15        11
## 6 Evangelical Prot  575       869      1064      982
## 7 Hindu            1         9         7         9
## 8 Historically Black Prot  228       244       236       238
## 9 Jehovah's Witness  20         27        24        24
## 10 Jewish           19         19        25        25
## 11 Mainline Prot    289       495       619       655
## 12 Mormon           29         40        48        51
## ...
```

Package `tidyverse`: basic operations (cont.)

```
relig_income %>%
  pivot_longer(-religion, names_to = "income", values_to = "count")

## # A tibble: 72 x 3
##       religion income   count
##       <chr>     <chr>    <dbl>
## 1 Agnostic  <$10k      27
## 2 Agnostic  $10-20k    34
## 3 Agnostic  $20-30k    60
## 4 Agnostic  $30-40k    81
## 5 Atheist    <$10k      12
## 6 Atheist    $10-20k    27
## 7 Atheist    $20-30k    37
...
....
```

Package `tidyverse`: basic operations (cont.)

Dataset: 2017 American Community Survey with median yearly income and median monthly rent estimates and margin of error

```
us_rent_income
```

```
## # A tibble: 104 x 5
##   GEOID NAME     variable estimate    moe
##   <chr> <chr>     <chr>      <dbl> <dbl>
## 1 01   Alabama   income     24476   136
## 2 01   Alabama   rent       747     3
## 3 02   Alaska    income     32940   508
## 4 02   Alaska    rent       1200    13
## 5 04   Arizona   income     27517   148
## 6 04   Arizona   rent       972     4
## 7 05   Arkansas  income     23789   165
## ...
```

Package tidyverse: basic operations (cont.)

```
us_rent_income %>%
  pivot_wider(names_from = variable, values_from = c(estimate, moe))

## # A tibble: 52 x 6
##   GEOID NAME           estimate_income estimate_rent moe_income moe_rent
##   <chr> <chr>            <dbl>        <dbl>       <dbl>      <dbl>
## 1 01    Alabama          24476         747       136
## 2 02    Alaska           32940        1200      508
## 3 04    Arizona          27517         972       148
## 4 05    Arkansas         23789         709       165
## 5 06    California       29454        1358      109
## 6 08    Colorado          32401        1125      109
## 7 09    Connecticut      35326        1123      195
## 8 10    Delaware          31560        1076      247
## 9 11    District of Columbia 43198        1424      681
## 10 12   Florida           25952        1077      70
## # ... with 42 more rows
```

tidyverse: notation remarks

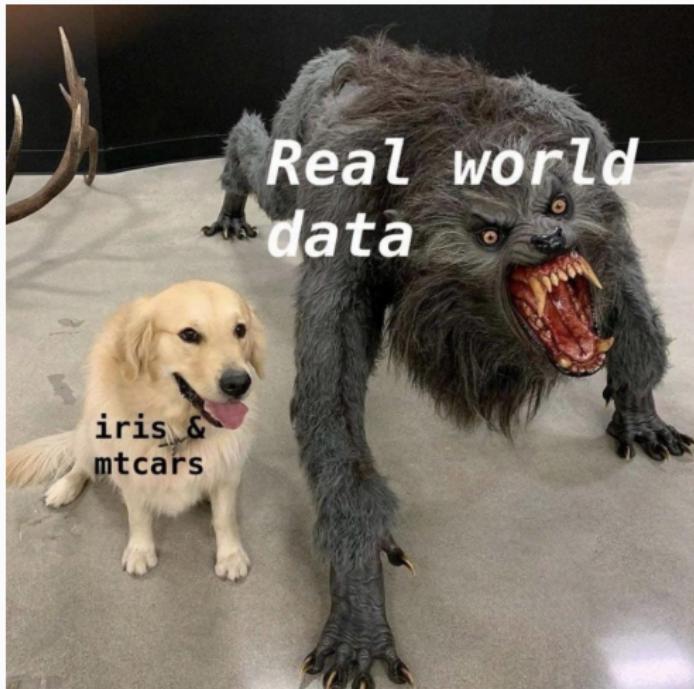
- $\%>\%$ is the chaining operator or pipe: $x \%>\% f(y)$ becomes $f(x, y)$
- $.$ represents the previous value in the chain, i.e. $x \%>\% f(.)$ becomes $f(x)$
- \sim is used for anonymous functions. i.e. `function(x) x + 2` can be written as
 - $\sim .x + 2$, where $.x$ represents the first argument of the function
 - $\sim . + 2$, in case the first argument is the previous value in the chain

R references for Data Wrangling

- R for Data Science, Hadley Wickham and Garrett Grolemund (2017)
- More details on these packages from tidyverse and other packages: RStudio Cheatsheets

Data Quality

Why?



Data Quality

- The raw format of real data is usually widely variable as values may be missing, inconsistent across different data sources, erroneous.
- Poor data quality poses several challenges to the effective data analysis

Example:

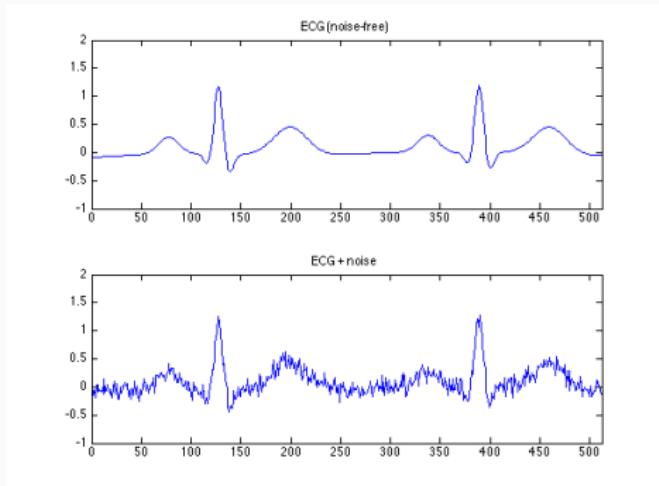
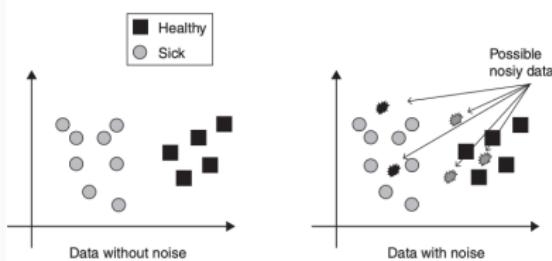
- A classification model for predicting a client's loan risks is built using poor data
 - credit-worthy candidates are denied loans
 - loans are given to individuals that default

Data Quality (cont.)

- What are the kinds of data quality problems?
 - How can we detect problems with the data?
 - What can we do about these problems?
-
- Examples of data quality problems:
 - Noise and outliers
 - Missing values
 - Duplicate data
 - Inconsistent or incorrect data

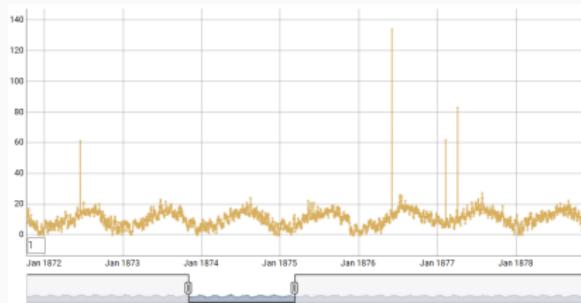
Noise

- Noise may refer to irrelevant or useless information
- It can be caused by incorrect or distorted measurements
- It can also be caused by the proper variability of the domain



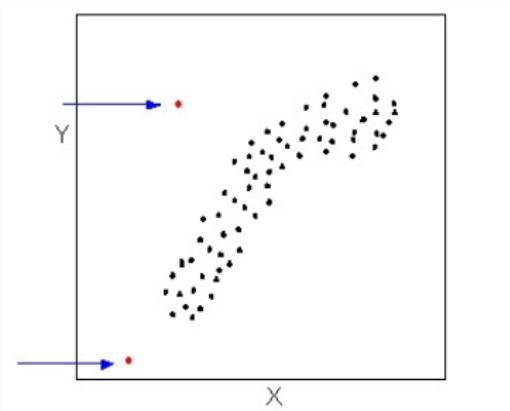
Outliers

- Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set
- Case 1: **outliers are noise** that interferes with data analysis
 - 130°C value for air temperature



Outliers (cont.)

- Case 2: outliers are the goal of our analysis
 - credit card fraud, intrusion detection



- What are the causes?

Missing Values

- Missing Completely at Random (MCAR)
 - missing value is independent of observed and unobserved data
 - there is nothing systematic about it
 - e.g. a lab value because a lab sample was processed improperly
- Missing at Random (MAR)
 - missing value is related to observed data, not to unobserved data.
 - there may be something systematic about it
 - e.g. missing income value may depend on the age
- Missing Not at Random (MNAR)
 - missing value is related to unobserved data of the variable itself
 - informative / non-ignorable missingness
 - e.g. a person did not enter his/her weight in a survey

Missing Values (cont.)

Solutions:

- remove observations with missing values, i.e. consider only complete cases
 - critical if there are many observations with missing values
- ignore missing values in the analytical phase
 - use methods that are inherently designed to work robustly with missing values
- make estimates to fill the missing values - imputation
 - the most common value of the attribute (e.g. mean, mode); based on other(s) attribute(s); more sophisticated methods
 - it might introduce bias in data and affect the results

Duplicates

- Data set may include data objects that are duplicates, or almost duplicates of one another
 - Major issue when merging data from heterogeneous sources
- Examples
 - Same person with multiple email addresses
- It is necessary a process of dealing with duplicate data issues
 - When should duplicate data not be removed?

Inconsistent or Incorrect Data

- This is the hardest type of data quality issues to detect
- It may depend on expert domain knowledge
- Examples:
 - 4/11/2000: Nov. 4th or April, 11th?
 - author name in a publication (e.g. John Smith, J. Smith, Smith J.)
 - a city called Shanghai in the United States

References

References

- Aggarwal, Charu C. 2015. *Data Mining, the Texbook*. Springer.
- Gama, João, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. 2015. *Extração de Conhecimento de Dados: Data Mining -3rd Edition*. Edições Sílabo.
- Gandomi, Amir, and Murtaza Haider. 2015. "Beyond the Hype: Big Data Concepts, Methods, and Analytics." *International Journal of Information Management* 35 (2): 137–44. doi:<https://doi.org/10.1016/j.ijinfomgt.2014.10.007>.
- Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Moreira, João, Andre Carvalho, and Tomás Horvath. 2018. *Data Analytics: A General Introduction*. Wiley.
- "R Project." 2021. <https://www.r-project.org/>.
- Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining*. 2nd ed. Pearson.

Data Understanding and Preparation

Rita P. Ribeiro

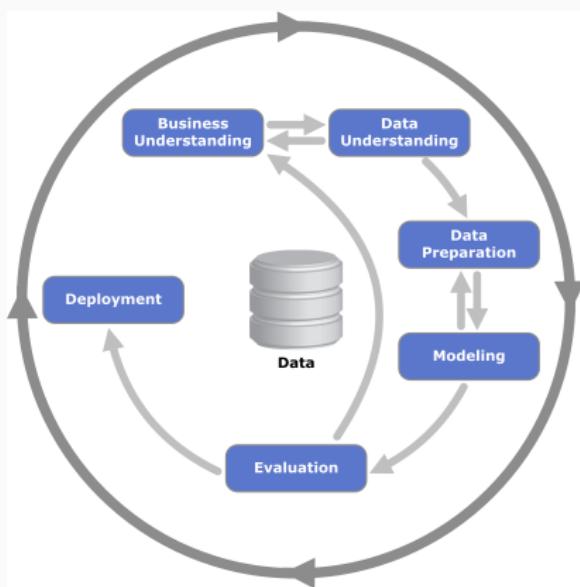
Machine Learning - 2021/2022



DEPARTAMENTO DE CIÉNCIA DE COMPUTADORES
FACULDADE DE CIÉNCIAS DA UNIVERSIDADE DO PORTO

CRISP-DM: a Typical Data Mining Workflow

- Cross-Industry Process for Data Mining (CRISP-DM)



Shearer C.: The CRISP-DM model: the new blueprint for data mining, J Data Warehousing (2000); 5:13—22.

CRISP-DM: Data Understanding



- Collect Initial Data:
initial data collection report
- Describe Data:
data description report
- Explore Data:
data exploration report
- Verify Data Quality:
data quality report

CRISP-DM: Data Preparation



- **Data Set:**
data set description
- **Select Data:**
rationale for inclusion/exclusion
- **Clean Data:**
data cleaning report
- **Construct Data:**
derived variables, generated records
- **Integrate Data:**
merged data
- **Format Data:**
reformatted data

Summary

- Data Understanding
 - Data Quality
 - Data Summarization
 - Data Visualization
- Data Preparation
 - Feature Extraction
 - Data Cleaning
 - Data Transformation
 - Feature Engineering
 - Data and Dimensionality Reduction

Data Understanding

Data Summarization

- Motivation
 - With big data sets it is hard to have an idea of what is going on in the data
 - Data summaries provide overviews of key properties of the data
 - Help selecting the most suitable tool for the analysis
 - Their goal is to describe important properties of the distribution of the values
- Types of Summaries
 - What is the “most common value”?
 - What is the “variability” in the values?
 - Are there “strange” / unexpected values in the data set?

Data Summarization (cont.)

- Data set
 - Univariate data
 - Multivariate data
- Variables
 - Categorical variables
 - Numeric variables

Data Summarization (cont.)

Example Data set

- `algae` data set composed by 200 water samples taken at several European rivers, which are described by:
 - 3 categorical variables: season, size and speed of the river
 - 8 numeric variables with chemical concentration measurements
 - 7 numeric variables with the concentration level of harmful algae.

Data Summarization: Categorical Variables

- Mode: the most frequent value
- Frequency table: frequency of each value (absolute or relative)

- season

autumn	spring	summer	winter
40	53	45	62

- Contingency tables: cross-frequency of values for two variables

- season and size

	autumn	spring	summer	winter
large	11	12	10	12
medium	16	21	21	26
small	13	20	14	24

Data Summarization: Numeric Variables

Statistics of location

- Mean (or sample mean) - sensitive to extreme values

$$\mu_x = \frac{1}{n} \sum_{i=1}^n x_i$$

- Median

- It is the 50^{th} -precentile, i.e. the value above (below) which there are 50% of the values in the data set

- Mode

- It is the most common (more frequently occurring) value in a set of values
 - Note that the mode can be applied to categorical variables

Data Summarization: Numeric Variables (cont.)

Statistics of variability or dispersion

- Range: $\max_x - \min_x$
- Variance σ_x^2 - sensitive to extreme values
- Standard Deviation - sensitive to extreme values

$$\sigma_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_x)^2}$$

- Inter-quartile Range (*IQR*)
 - It is the difference between the 3rd (Q_3) and 1st (Q_1) quartiles
 - Q_1 is the number below which there are 25% of the values
 - Q_3 is the number below which there are 75% of the values

Data Summarization: Numeric Variables (cont.)

"An outlier is a point that deviates so much from the other data points as to arouse suspicions that it was generated by a different mechanism" (Hawkins, 1980)

- For a numeric variable an outlier can be an extreme value
- In the presence of such values,
 - median or mode are more robust as a central tendency statistic
 - inter-quartile range is more appropriate as variability statistic.
- Boxplot definition (Tukey, 1977)
 - any value outside the interval $[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$ is an outlier

Data Summarization: Numeric Variables (cont.)

Multivariate analysis of variability or dispersion

- Covariance Matrix: variance between every pair of numeric variables - the value depends on the magnitude of the variable

$$cov(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)$$

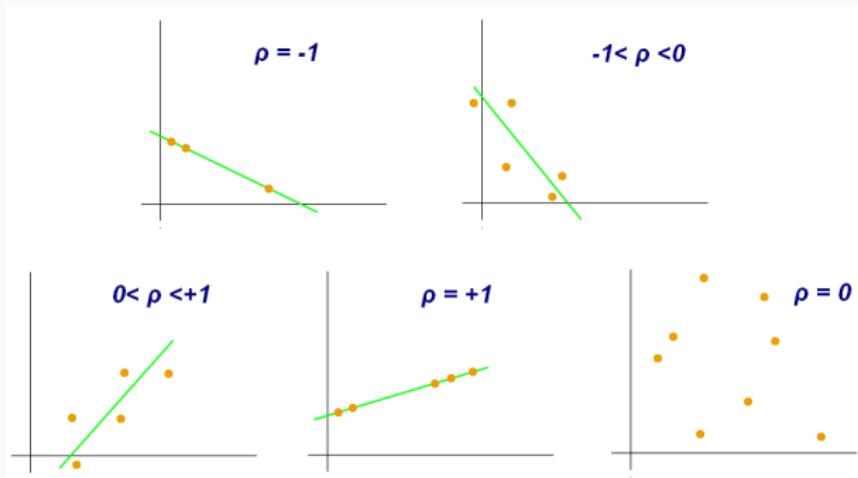
- Correlation Matrix: correlation between every pair of numeric variables - the influence of the magnitude is removed

$$cor(x, y) = \frac{cov(x, y)}{\sigma_x \sigma_y}$$

Data Summarization: Numeric Variables (cont.)

Pearson Correlation Coefficient (ρ):

- measures the linear correlation between two variables;
- it has a value between +1 and -1.



Data Summarization: Numeric Variables (cont.)

Pearson Correlation Coefficient - cont.

For a given sample of two variables x and y , $\{(x_1, y_1), \dots, (x_n, y_n)\}$, the correlation coefficient is defined as

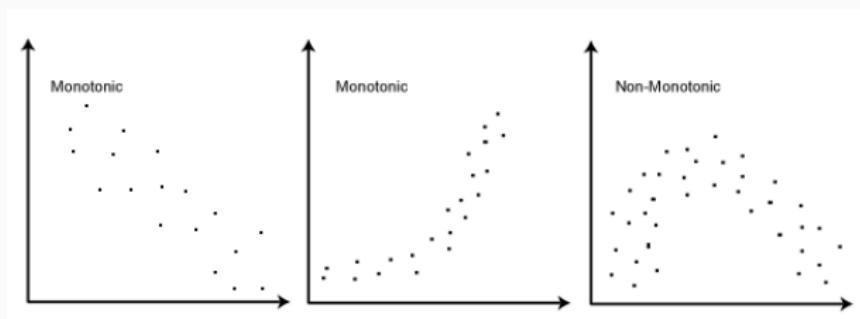
$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

where n is the sample size, x_i and y_i are the individual sample points and $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ is the sample mean, the same for \bar{y}

Data Summarization: Numeric Variables (cont.)

Spearman Rank-Order Correlation Coefficient:

- measures the strength and direction of monotonic association between two variables;
- two variables can be related according to a type of non-linear but still monotonic relationship.



Data Summarization: Numeric Variables (cont.)

Spearman Rank-Order Correlation Coefficient: cont.

- a rank-based, and non-parametric, version of *Pearson* correlation coefficient;
- it has a value between +1 and -1;

$$rs_{xy} = r_{rank_x rank_y}$$

- if all n ranks are distinct integers, it can be computed using the popular formula

$$rs_{xy} = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}$$

where $d_i = rank_{x_i} - rank_{y_i}$ is the difference between the two ranks of each observation.

Data Visualization

Data Visualization

- Motivation
 - Humans are outstanding at detecting patterns and structures with their eyes
 - Data visualization methods try to explore these capabilities
 - Help detecting patterns and trends, and also outliers and unusual patterns
- Main Types of Graphs
 - Univariate Graphs
 - Bivariate Graphs
 - Multivariate / Conditioned Graphs

Data Visualization: Univariate Graphs

- Categorical Variables

- Barplots
- Piecharts
- ...

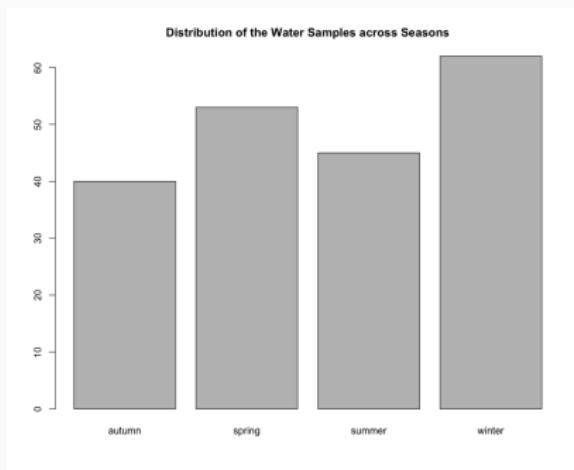
- Numeric Variables

- Line plots
- Histograms
- QQ Plots
- Boxplots
- ...

Data Visualization: Univariate Graphs (cont.)

Barplots

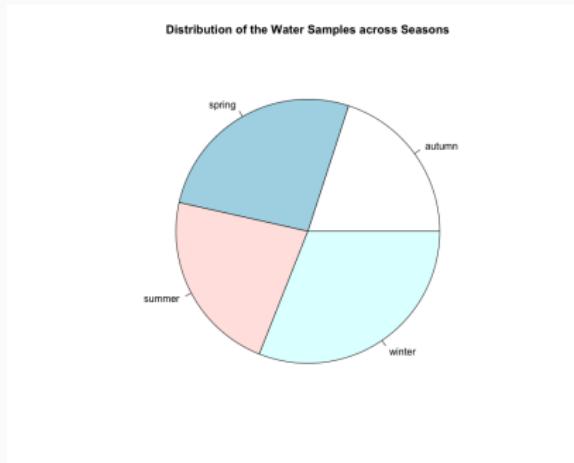
- The main purpose is to display a set of values as heights of bars
- It can be used to display the frequency of occurrence of different values of a categorical variable



Data Visualization: Univariate Graphs (cont.)

Piecharts

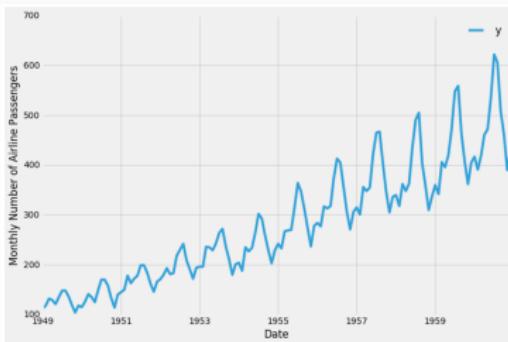
- Have the same purpose as bar plots but with information in the form of a pie.
- Are not so good for comparison purposes



Data Visualization: Univariate Graphs (cont.)

Line Plots

- The main purpose is to analyze the evolution of the values of a continuous variable.
- x-axis represent a quantitative scale with equal lag between observations.
- Frequently used to deal with the notion of time



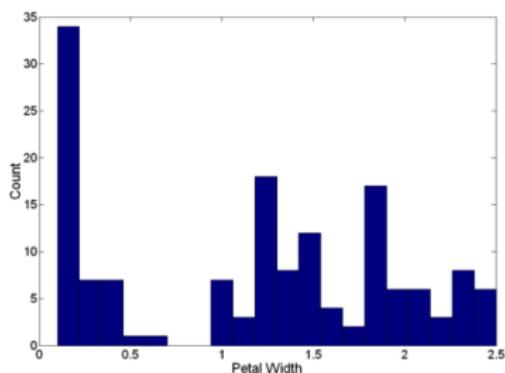
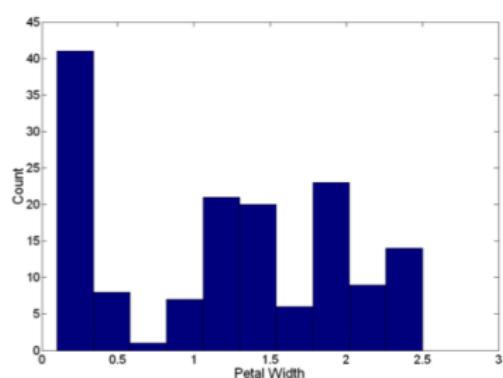
Histograms

- The main purpose is to display how the values of a continuous variable are distributed
- It is obtained as follows:
 - first, the range of the variable is divided into a set of bins (intervals of values)
 - then, the number of occurrences of values on each bin is counted
 - then, this number is displayed as a bar

Data Visualization: Univariate Graphs (cont.)

Problems with Histograms

- Histograms may be misleading in small data sets
- The shape of the histogram depends on the number of bins
- How are the limits of the bins chosen? There are several algorithms for this.



Data Visualization: Univariate Graphs (cont.)

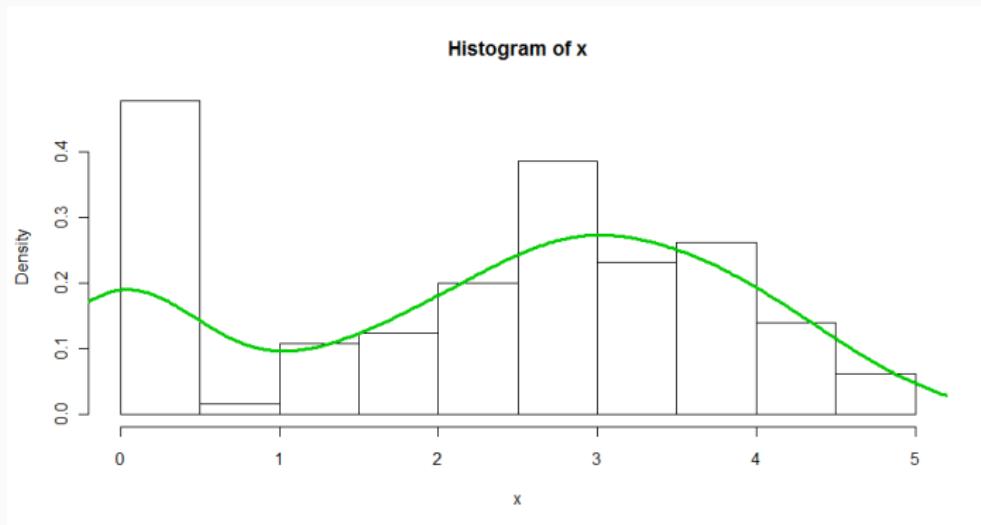
- Some of the problems of histograms can be tackled by smoothing the estimates of the distribution of the values. That is the purpose of kernel density estimates
- Kernel estimates calculate the estimate of the distribution at a certain point by smoothly averaging over the neighboring points
- Namely, the density is estimated by

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

- where $K(\cdot)$ is the kernel — a non-negative function — and $h > 0$ is a smoothing parameter called the bandwidth.

Data Visualization: Univariate Graphs (cont.)

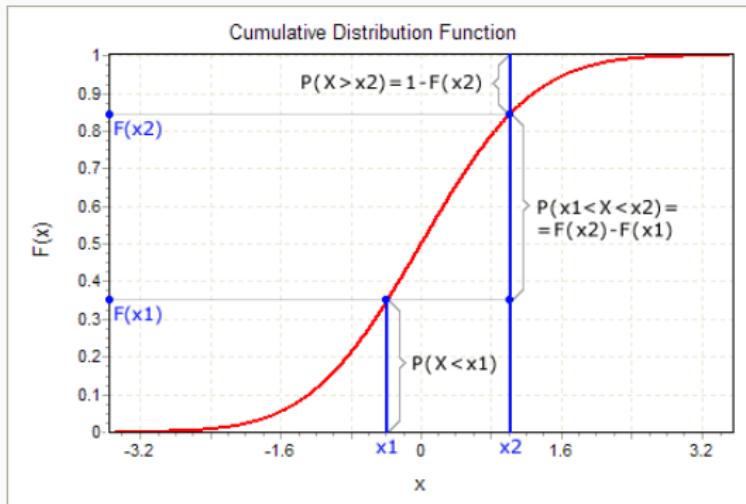
- Histogram with density estimate



Data Visualization: Univariate Graphs (cont.)

Cumulative Distribution Function (CDF)

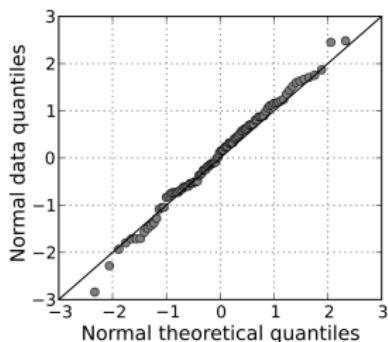
- CDF of a random variable X : $F_X(x) = P(X \leq x)$



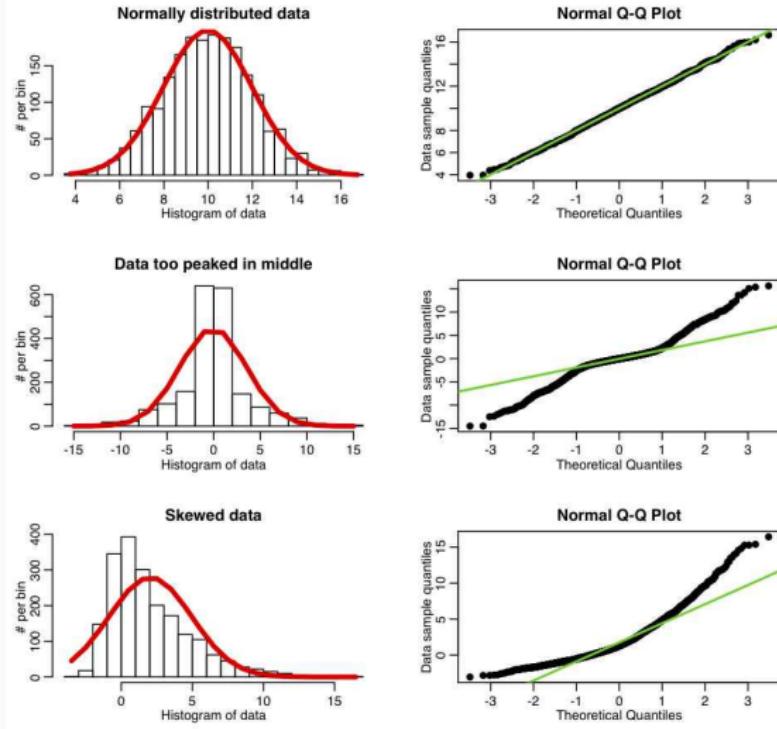
Data Visualization: Univariate Graphs (cont.)

QQ Plots

- A graphical view of how properties such as location, scale, and skewness compare in two distributions.
- Can be used to visually check the hypothesis that the variable under study follows a normal distribution, comparing the observed distribution against the Normal distribution.



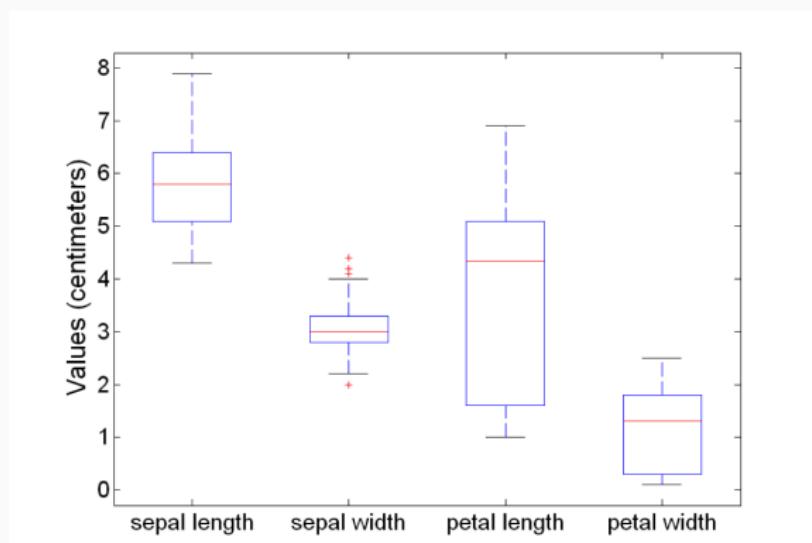
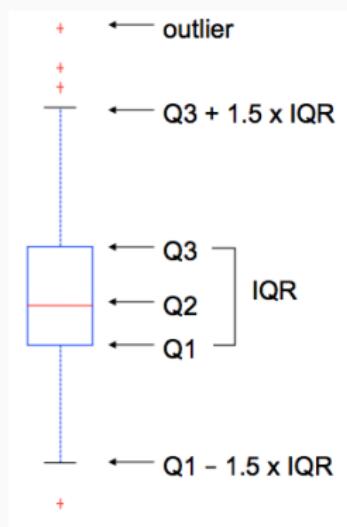
Data Visualization: Univariate Graphs (cont.)



Data Visualization: Univariate Graphs (cont.)

Boxplots

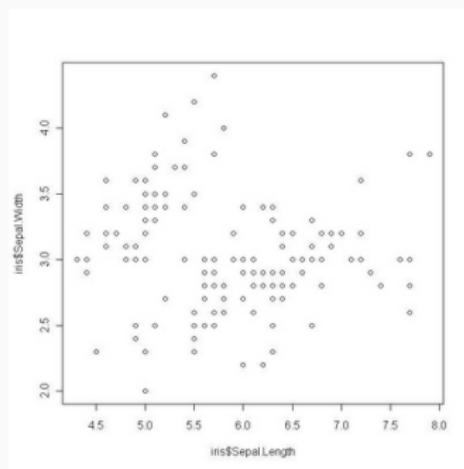
- Box plot provide an interesting summary of a variable distribution
- For instance, they inform us of the interquartile range and of the outliers (if any)



Data Visualization: Bivariate Graphs

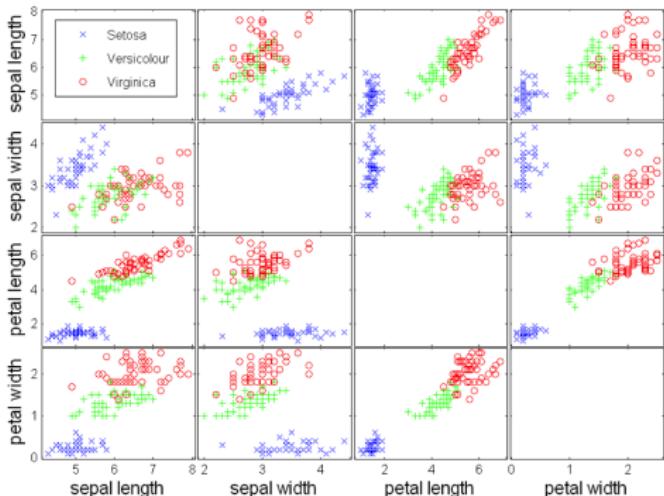
Scatterplots

- The natural graph for showing the relationship between two numeric variables



Data Visualization: Multivariate Graphs

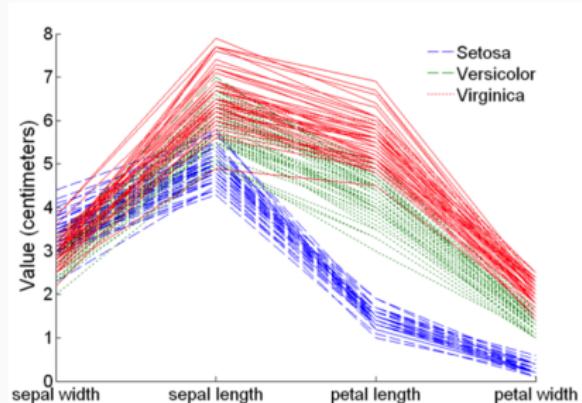
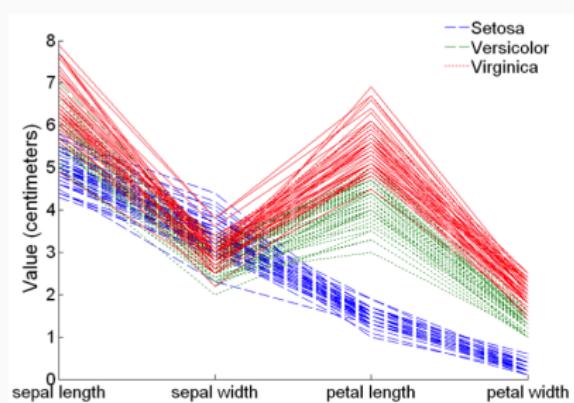
- The scatterplot can plot the relationship between every pair of numeric variables and respective groups



Data Visualization: Multivariate Graphs (cont.)

Parallel Coordinates Plot

- Plots attributes values for each case (represented as a line)
- The order might be important to help identifying groups



Data Visualization: Multivariate Graphs (cont.)

Correlogram

- Chart of correlation statistics (e.g. pearson) for each pair of variables.



Conditioned Graphs

- Data sets frequently have categorical variables, which values can be used to create sub-groups of the data.
 - e.g. the sub-group of male clients of a company
- Conditioned plots allow the simultaneous presentation of these sub-group graphs to better allow finding eventual differences between the sub-groups
 - Conditioned Histograms
 - Conditioned Boxplots
 - ...

The Grammar of Graphics in R

The Grammar of Graphics in R: `ggplot2`

- Package `ggplot2` implements the ideas created by Wilkinson (2005) on a grammar of graphics
- This grammar is the result of a theoretical study on what is a statistical graphic
- `ggplot2` builds upon this theory by implementing the concept of a layered grammar of graphics (Wickham, 2009)
- The grammar defines a statistical graphic as:
 - a mapping from data into **aesthetic attributes** (color, shape, size, etc.) of **geometric objects** (points, lines, bars, etc.)

The Grammar of Graphics in R: ggplot2 (cont.)

- Main idea: specify the layers that make up the graphic, independently, and add them together with **+**
- Key elements of a statistical graphic:
 - data
 - aesthetic mappings
 - geometric objects
 - statistical transformations
 - scales
 - coordinate system
 - faceting
 - labelling

The Grammar of Graphics in R: ggplot2 (cont.)

Aesthetic Mappings

- Controls the relation between **data variables** and **graphic variables**
 - e.g., map the Temperature variable of a data set into the *x*-axis in a scatter plot
 - e.g., map the Species of a plant into the *color* of dots in a graphic
- Some examples
 - position: `aes(x=..., y=...)`
 - color: `aes(color=...)`
 - fill: `aes(fill=...)`
 - shape: `aes(shape=...)`
 - linetype: `aes(linetype=...)`
 - size: `aes(size=...)`

The Grammar of Graphics in R: ggplot2 (cont.)

Geometric Objects

- Controls what is shown in the graphics
 - e.g., show each observation by a point using the aesthetic mappings that relate two variables in the data set into the x -axis, y -axis of the graphic
- Some Examples:
 - scatterplot: `geom_point()`
 - line plot: `geom_line()`
 - boxplot: `geom_boxplot()`
 - histogram: `geom_histogram()`
 - barplot: `geom_bar()`

The Grammar of Graphics in R: ggplot2 (cont.)

Statistical Transformations

- Calculates and performs statistical analysis over the data in the graphic
 - e.g., count occurrences of certain values
 - e.g., discretize by creating bins
 - e.g., calculate the density by a density estimation function
- Some Examples:
 - `stat_count(geom="bar") / geom_bar(stat="count")`
 - `stat_bin(geom="bar") / geom_histogram(stat="bin")`
 - `stat_density(geom="area") / geom_area(stat="density")`
 - `...count..., ...density...`: variables created by the statistic

The Grammar of Graphics in R: ggplot2 (cont.)

Scales

- Maps the data values into values in the coordinate system of the graphics device

Scale	Types	Examples
<code>scale_color_</code>	identity	<code>scale_color_discrete()</code>
<code>scale_fill_</code>	manual	<code>scale_fill_continuous()</code>
<code>scale_size_</code>	continuous	<code>scale_size_manual()</code>
	discrete	<code>scale_size_discrete()</code>
<code>scale_shape_</code>	discrete	<code>scale_shape_discrete()</code>
<code>scale_linetype_</code>	identity	<code>scale_shape_manual()</code>
	manual	<code>scale_linetype_discrete()</code>
<code>scale_x_</code>	continuous	<code>scale_x_continuous()</code>
<code>scale_y_</code>	discrete	<code>scale_y_discrete()</code>
	reverse	<code>scale_x_reverse()</code>
	log	<code>scale_y_log()</code>
	date	<code>scale_x_date()</code>
	datetime	<code>scale_y_datetime()</code>

The Grammar of Graphics in R: ggplot2 (cont.)

Coordinate System

- The coordinate system used to plot the data
- Some Examples:
 - Cartesian: `coord_cartesian()`
 - Polar: `coord_polar()`

Faceting

- Split the data into sub-groups and draw sub-graphs for each group (Conditioned Graphs)
- Examples:
 - `facet_wrap()`: defines groups according to the nominal values of a categorical variable
 - `facet_grid()`: defines groups according to the crossing of nominal values of two categorical variables

The Grammar of Graphics in R: ggplot2 (cont.)

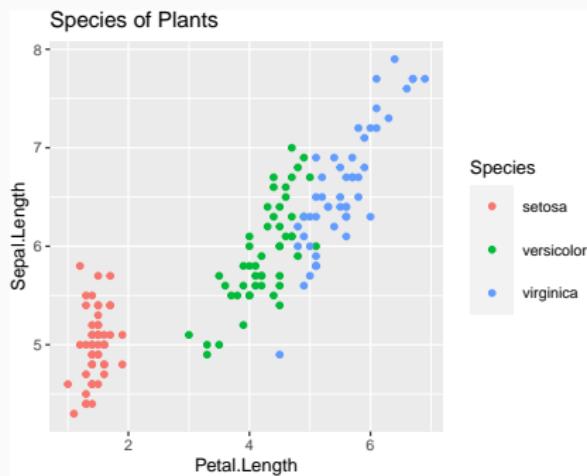
Labelling

- Label x -axis, y -axis, title of the graphic
- Some examples:
 - `ggtitle()`
 - `xlab()`
 - `ylab()`
 - `labs(title=..., x=..., y=...)`

The Grammar of Graphics in R: ggplot2 (cont.)

Example 1: scatterplot

```
ggplot(iris, aes(x = Petal.Length, y = Sepal.Length,  
color = Species)) + geom_point() + ggtitle("Species of Plants")
```



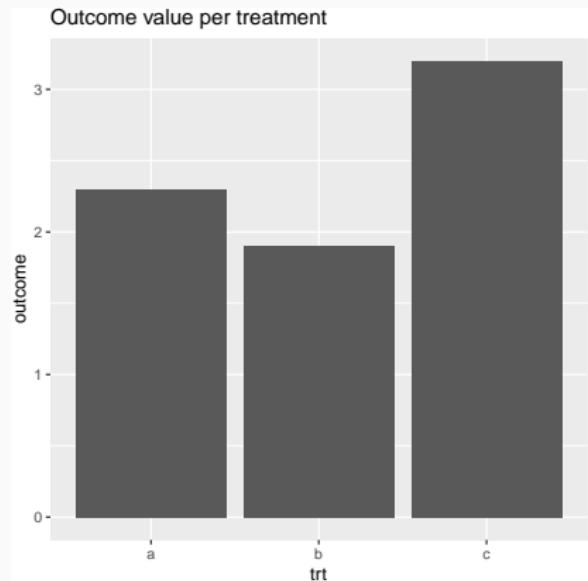
The Grammar of Graphics in R: ggplot2 (cont.)

- There are two types of bar charts
 - `geom_col()`
 - makes the height of the bar representing values in the data
 - `geom_bar()`
 - makes the height of the bar proportional to the number of cases in each group

The Grammar of Graphics in R: ggplot2 (cont.)

Example 2: barplot - I

```
ggplot(df, aes(x = trt, y = outcome)) + geom_col() +  
  ggtitle("Outcome value per treatment")
```

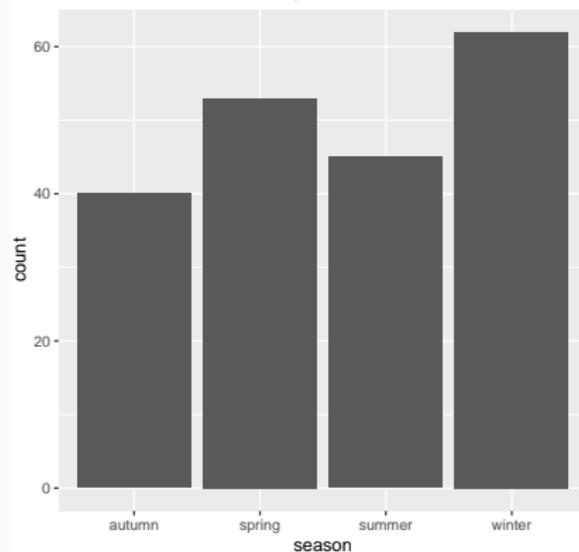


The Grammar of Graphics in R: ggplot2 (cont.)

Example 3: barplot - II

```
ggplot(algae, aes(x = season)) + geom_bar() +  
  ggtitle("Distribution of water samples across seasons")
```

Distribution of water samples across seasons

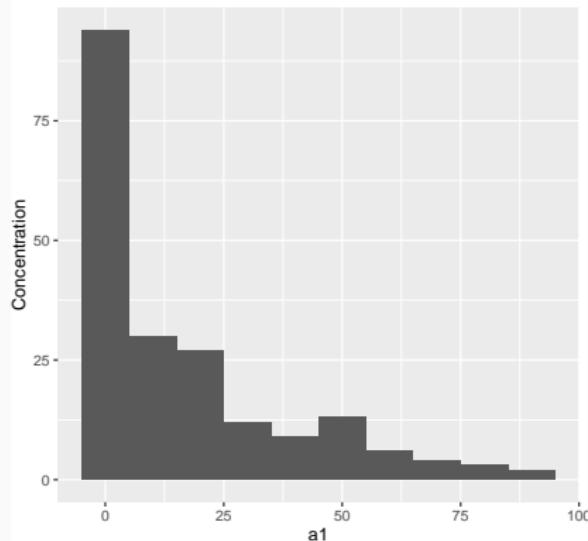


The Grammar of Graphics in R: ggplot2 (cont.)

Example 4: histogram

```
ggplot(algae, aes(x = a1)) + geom_histogram(binwidth = 10) +
  ggtitle("Distribution of Algae a1") + ylab("Concentration")
```

Distribution of Algae a1

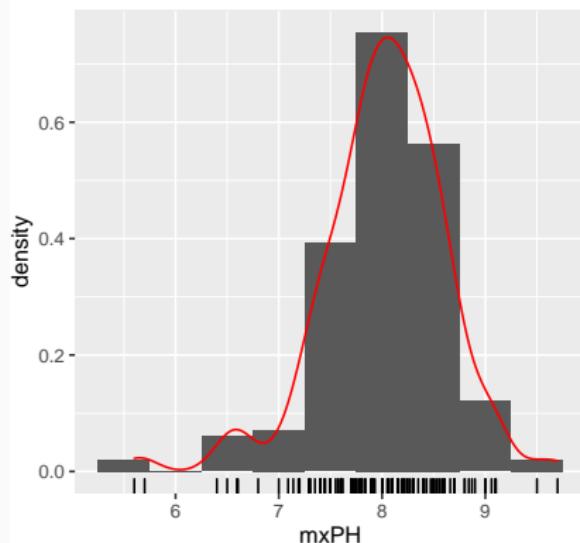


The Grammar of Graphics in R: ggplot2 (cont.)

Example 5: histogram with density estimation

```
ggplot(algae, aes(x = mxPH)) + geom_histogram(binwidth = 0.5,  
aes(y = ..density..)) + geom_density(color = "red") + geom_rug() +  
ggttitle("The Histogram of mxPH")
```

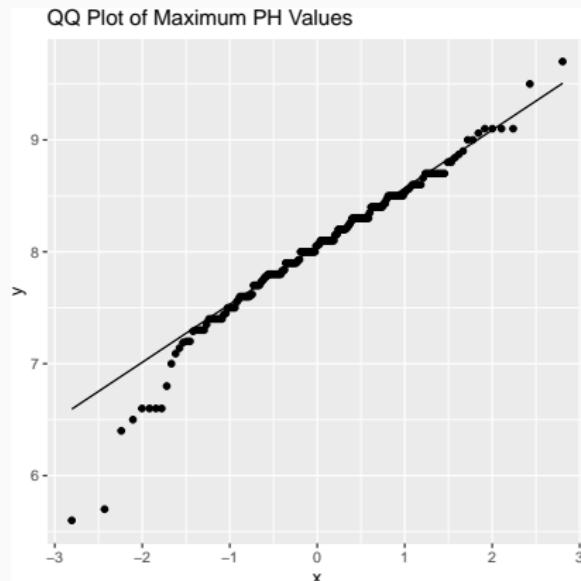
The Histogram of mxPH



The Grammar of Graphics in R: ggplot2 (cont.)

Example 6: QQ plot

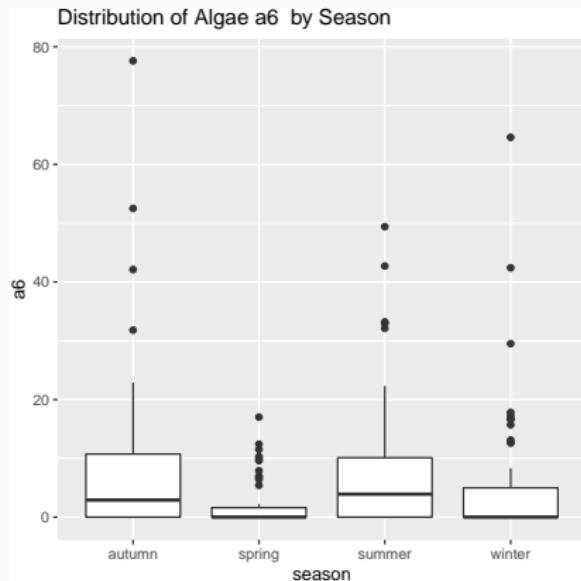
```
ggplot(algae, aes(sample = mxPH)) + geom_qq(geom = "point") +  
  stat_qq_line() + ggtitle("QQ Plot of Maximum PH Values")
```



The Grammar of Graphics in R: ggplot2 (cont.)

Example 7: conditioned boxplot

```
ggplot(algae, aes(x = season, y = a6)) +  
  geom_boxplot() + ggtitle("Distribution of Algae a6 by Season")
```

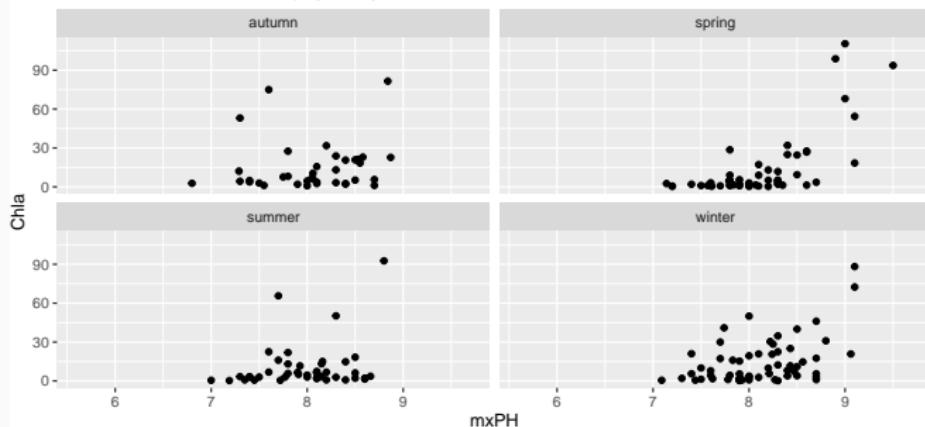


The Grammar of Graphics in R: ggplot2 (cont.)

Example 8: conditioned scatterplot

```
ggplot(algae, aes(x = mxPH, y = Chla)) +  
  geom_point() + facet_wrap(~season) +  
  ggtitle("Maximum PH and Chlorophyll a by Season")
```

Maximum PH and Chlorophyll a by Season

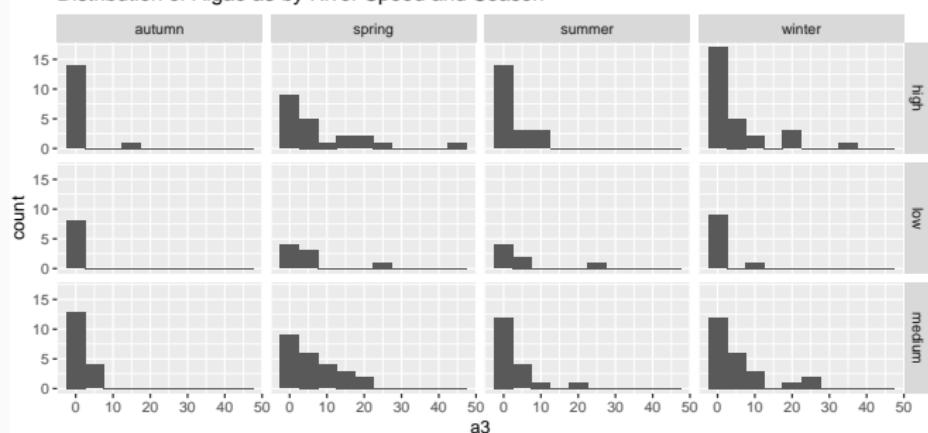


The Grammar of Graphics in R: ggplot2 (cont.)

Example 9: conditioned histogram

```
ggplot(algae, aes(x = a3)) + geom_histogram(binwidth = 5) +  
  facet_grid(speed ~ season) + ggtitle("Distribution of Algae a3 by River Spee
```

Distribution of Algae a3 by River Speed and Season



The Grammar of Graphics in R: ggplot2 (cont.)

- Many more interesting things can be done with ggplot2
- For a more complete reference
 - R Graphics Cookbook, 2nd edition

Data Preparation

Data Preparation

Set of steps that may be necessary to carry out before any further analysis takes place on the available data.

- Data can come from a multitude of different sources
- Frequently, we have data sets with unknown variable values
- Many data mining methods are sensitive to the scale and/or the type of variables
 - Different variables may have different scales
 - Some methods are unable to handle either nominal or numerical variables

Data Preparation (cont.)

- We may face the need to “create” new variables to achieve our objectives
 - Sometimes we are more interested in relative values (variations) than absolute values
 - We may be aware of some domain-specific mathematical relationship among two or more variables that is important for the task
- Our data set may be too large for some methods to be applicable

Data Preparation (cont.)

- Feature Extraction
 - extract features from raw data on which analysis can be performed.
- Data Cleaning
 - data may be hard to read or require extra parsing efforts.
- Data Transformation
 - it may be necessary to change some of the values of the data.
- Feature Engineering
 - to incorporate some domain knowledge.
- Data and Dimensionality Reduction
 - to make modeling possible.

Feature Extraction

- It is very application specific and a very crucial step.
 - **sensor data:** large volume of low-level signals associated with date/time attributes
 - **image data:** very high-dimensional data that can be represented by pixels, color histograms, etc.
 - **web logs:** text in a prespecified format with both categorical and numerical attributes
 - **network traffic:** network packets information
 - **document data:** raw and unstructured data

Data Cleaning: Handling Missing Values

Ultimate Goal

- Making our data set `tidy`
 - each value belongs to a variable and an observation
 - each variable contains all values of a certain property measured across all observations
 - each observation contains all values of the variables measured for the respective case
- These properties lead to data tables where:
 - each row represents an observation
 - each column represents an attribute measured for each observation

Data Cleaning: Handling Missing Values (cont.)

Main Strategies

- Remove all cases in a data set with some unknown value
- Fill-in the unknowns with the imputation of the most common value (a statistic of centrality)
- Fill-in with the most common value on the cases that are more “similar” to the one with unknowns.
- Fill-in with linear interpolation of nearby values in time and/or space.
- Explore eventual correlations between variables
- Do nothing: many data mining methods are designated to work robustly with missing values

Data Cleaning: Handling Incorrect Values

- Inconsistency detection
 - data integration techniques within the database field
- Domain knowledge
 - data auditing that use domain knowledge and constraints
- Data-centric methods
 - statistical-based methods to detect outliers

Data Transformation

- Map entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
- Why it may be useful?
 - Imagine two attributes (e.g. age and salary) with a very different scale
 - Any aggregation function (e.g. euclidean distance) computed on the set of cases, will be dominated by the attribute of larger magnitude.
- Some common strategies:
 - Normalization
 - Binarization / One-Hot Encoding
 - Discretization

Data Transformation: Normalization

- Min-Max Scaling (Range-based Normalization)

$$y_i = \frac{x_i - \min_x}{\max_x - \min_x}$$

- \min_x and \max_x are the minimum and maximum values of attribute x
- values will lie in the range $[0, 1]$
- It is not robust for scenarios where there are outliers**
 - if an erroneous age value of 800 is registered instead of 80, most of the values will be in the range $[0, 0.1]$

Data Transformation: Normalization (cont.)

- Standardization (z-score Normalization):

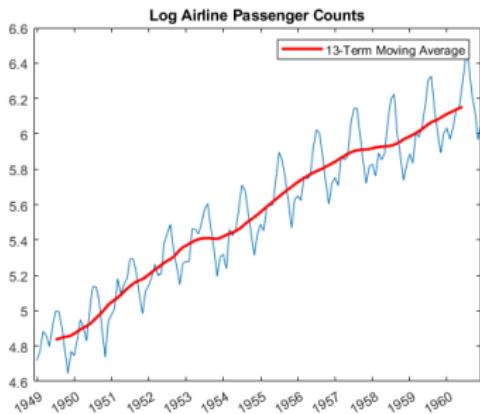
$$y_i = \frac{x_i - \mu_x}{\sigma_x}$$

- μ_x and σ_x are the mean and the standard deviation of attribute x
- values are rescaled so that they have $\mu_x = 0$ and $\sigma_x = 1$
- values will, typically, lie in the range $[-3, 3]$ under a normal distribution assumption

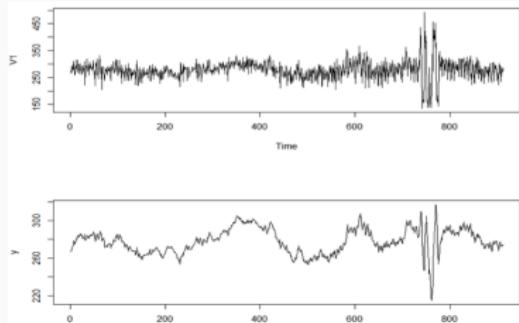
Data Transformation: Case Dependencies

- In time series it is common to use different techniques.
- Examples:
 - to adjust mean, variance, range
 - to remove unwanted, common signal

Moving Average



Low-pass filter



Data Transformation: Binarization / One-Hot Encoding

- Some data mining methods are only able to handle numeric attributes.
- If the categorical attribute is not ordinal, it is necessary to convert it into a numerical attribute.
- **Binarization:** if the attribute has only 2 possible nominal values, it can be transformed into 1 binary attribute
 - fever: yes/no – > fever: 1/0
- **One-Hot Encoding:** if the attribute has k possible nominal values, it can be transformed into k binary attributes
 - eye_color: brown/blue/green → eye_brown: 1/0, eye_blue: 1/0, eye_green: 1/0

Data Transformation: Discretization

- Process of converting a continuous attribute into an ordinal attribute of numeric variables.
- Some unsupervised discretization: find breaks in the data values
 - Equal-width
 - it divides the original values into equal-width range of values
 - it may be affected by the presence of outliers
 - Equal-frequency
 - it divides the original values so that the same number of values are assigned to each range
 - it can generate ranges with very different amplitudes
- Supervised discretization: use class labels to find breaks (we'll see later)

Feature Engineering

Fundamental to the application of machine learning.

'(...) some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.' - Pedro Domingos, 2012

- The process of using domain knowledge of the data to create features that might help when solving the problem.
- New features that can capture the important information in a data set much more efficiently than the original features.
- Case 1: express known relationships between existing variables
 - create ratios and proportions like credit card sales per person
 - from web logs obtain the average session duration per user, the frequency of access, etc.

Feature Engineering: Cases Dependencies

- Case 2: overcome limitations of some data mining tools regarding cases dependencies.
 - some tools shuffle the cases, or are not able to use the information about their dependencies (time, space, space-time)
 - two main ways of handling this issue:
 - constrain ourselves to tools that handle these dependencies directly
 - create variables that express the dependency relationships
- In time series is common to create features that represent **relative values instead of absolute values**, so to avoid trend effects.

$$y_t = \frac{x_t - x_{t-1}}{x_{t-1}}$$

Feature Engineering: Cases Dependencies (cont.)

- Other common technique is Time Delay Embedding
- Create variables whose values are the value of the same variable in previous time steps
 - If we have variables whose values are the value of the same variable but on different time steps, standard tools will be able to model the time relationships with these embeddings
 - Note that similar “tricks” can be done with space and space-time dependencies

X_{t-3}	X_{t-2}	X_{t-1}	X_t
X_{t_1}	X_{t_2}	X_{t_3}	X_{t_4}
X_{t_2}	X_{t_3}	X_{t_4}	X_{t_5}
...			
$X_{t_{n-3}}$	$X_{t_{n-3}}$	$X_{t_{n-1}}$	X_{t_n}

References

References

- Aggarwal, Charu C. 2015. *Data Mining, the Texbook*. Springer.
- Gama, João, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. 2015. *Extração de Conhecimento de Dados: Data Mining -3rd Edition*. Edições Sílabo.
- Gandomi, Amir, and Murtaza Haider. 2015. "Beyond the Hype: Big Data Concepts, Methods, and Analytics." *International Journal of Information Management* 35 (2): 137–44. doi:<https://doi.org/10.1016/j.ijinfomgt.2014.10.007>.
- Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Moreira, João, Andre Carvalho, and Tomás Horvath. 2018. *Data Analytics: A General Introduction*. Wiley.
- "R Project." 2021. <https://www.r-project.org/>.
- Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining*. 2nd ed. Pearson.

advanced issues in data preparation and modeling

Carlos Soares

(partly using materials kindly provided by José Luís Borges, from Han, Kamber & Pei, from Moreira, Carvalho & Horvath and from Ceja)

plan

- advanced issues in data preparation
 - data reduction
- ... and modeling
 - dealing with unbalanced classes
- final reflections on data quality

- advanced issues in data preparation
 - data reduction
 - context
 - attribute aggregation
 - feature selection
- ... and modeling
 - dealing with unbalanced classes
- final reflections on data quality

data reduction

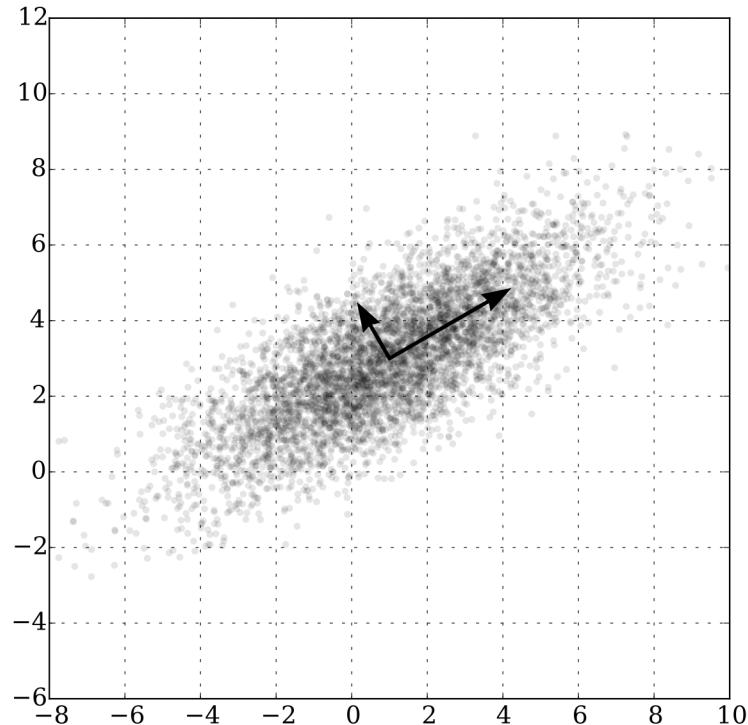
- obtain a reduced representation of the data set that is
 - much smaller in volume
- ... producing the **same analytical results**
 - or almost the same
- ... improved visualization of data
- ... with more interpretable models
- ... much faster

dimensionality reduction: data is cursed!

- curse of dimensionality
 - when dimensionality increases, data becomes increasingly sparse
 - density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
 - the possible combinations of subspaces will grow exponentially
- ... number of data points required for robust patterns grows exponentially with number of attributes

two approaches

attribute aggregation

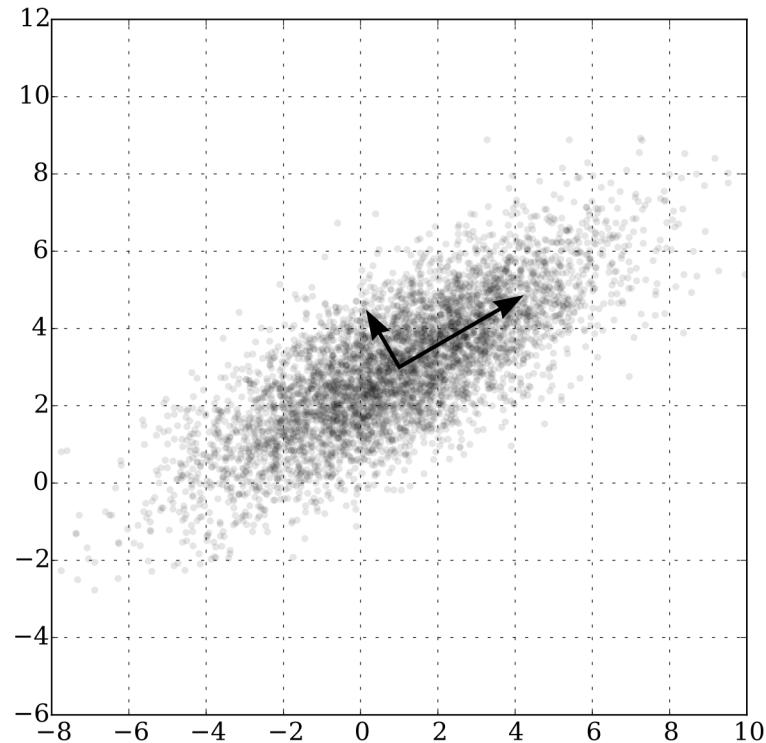


feature selection



attribute aggregation: PCA

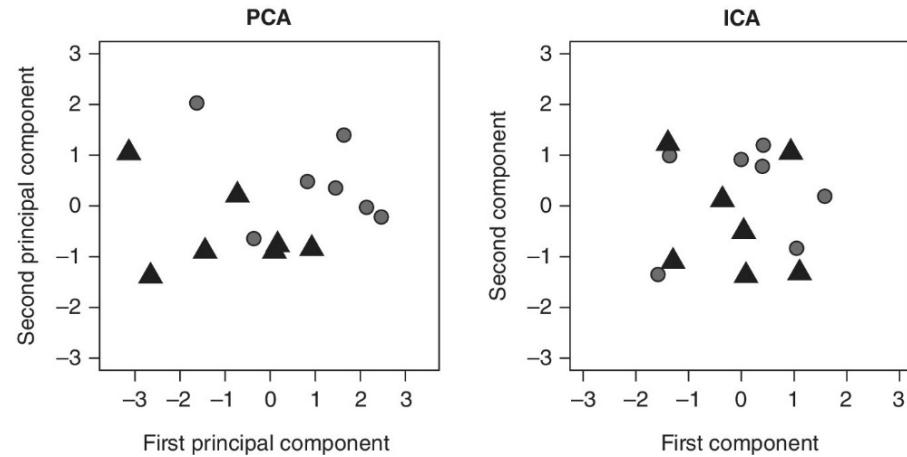
- Principal Component Analysis
 - n new features
 - linear combinations of existing n attributes
 - orthogonal to each other
 - $k \ll n$ explain most of the variance in the data



By Nicoguaro - Own work, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=46871195>

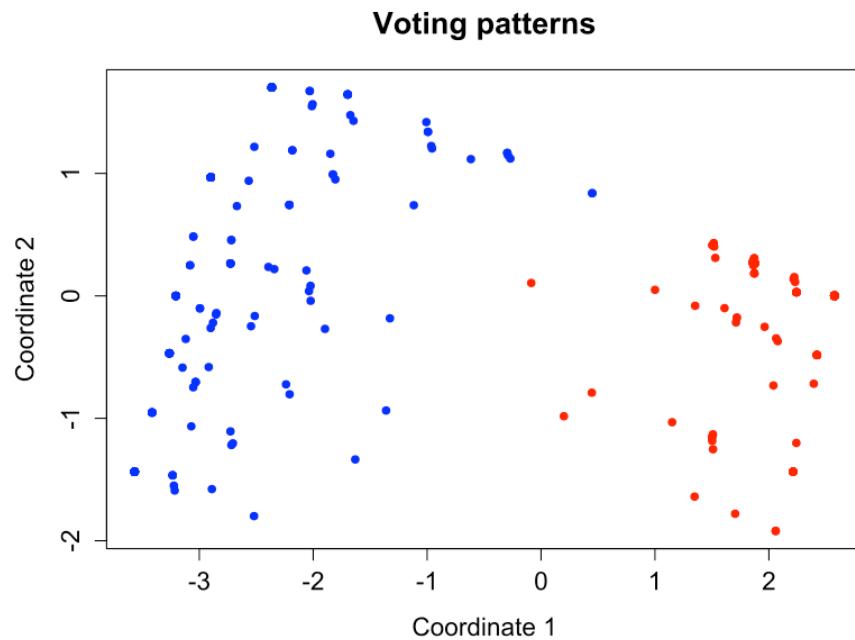
attribute aggregation: ICA vs PCA

- both create linear combinations of the attributes
- ICA assumes the original attributes are statistically independent
- ... reduces higher order statistics
 - e.g. kurtosis
- ... does not rank components



attribute aggregation: multidimensional scaling

- linear projection of a data set
- uses the distances between pairs of objects
 - not the values of the attributes of the objects
- particularly suitable when it is difficult to extract relevant features to represent the objects



https://en.wikipedia.org/wiki/Multidimensional_scaling

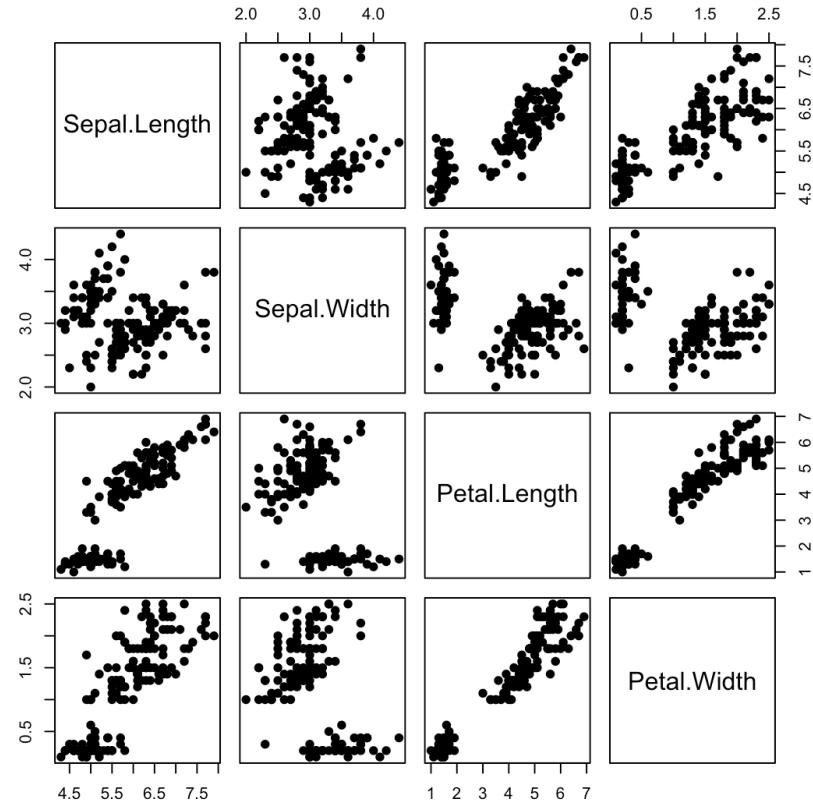
- advanced issues in data preparation
 - data reduction
 - context
 - attribute aggregation
 - feature selection
- ... and modeling
 - dealing with unbalanced classes
- final reflections on data quality

feature selection: eliminate...

- redundant attributes
 - duplicate much or all of the information contained in one or more other attributes
 - e.g. purchase price of a product and the amount of sales tax paid
- irrelevant attributes
 - contain no useful information
 - e.g. students' ID is often irrelevant to the task of predicting students' GPA

feature selection: filter methods

- 2 attributes
 - remove redundant attributes
- 1 attribute vs target
 - identify relevant variables



feature selection: wrapper methods (1/4)

Feature Selection

Full Feature Set



<https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96>

feature selection: wrapper methods (2/4)

Feature Selection

Full Feature Set



<https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96>

feature selection: wrapper methods (3/4)

Feature Selection

Full Feature Set



<https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96>

feature selection: wrapper methods (4/4)

Feature Selection

Full Feature Set



Identify Useful Features



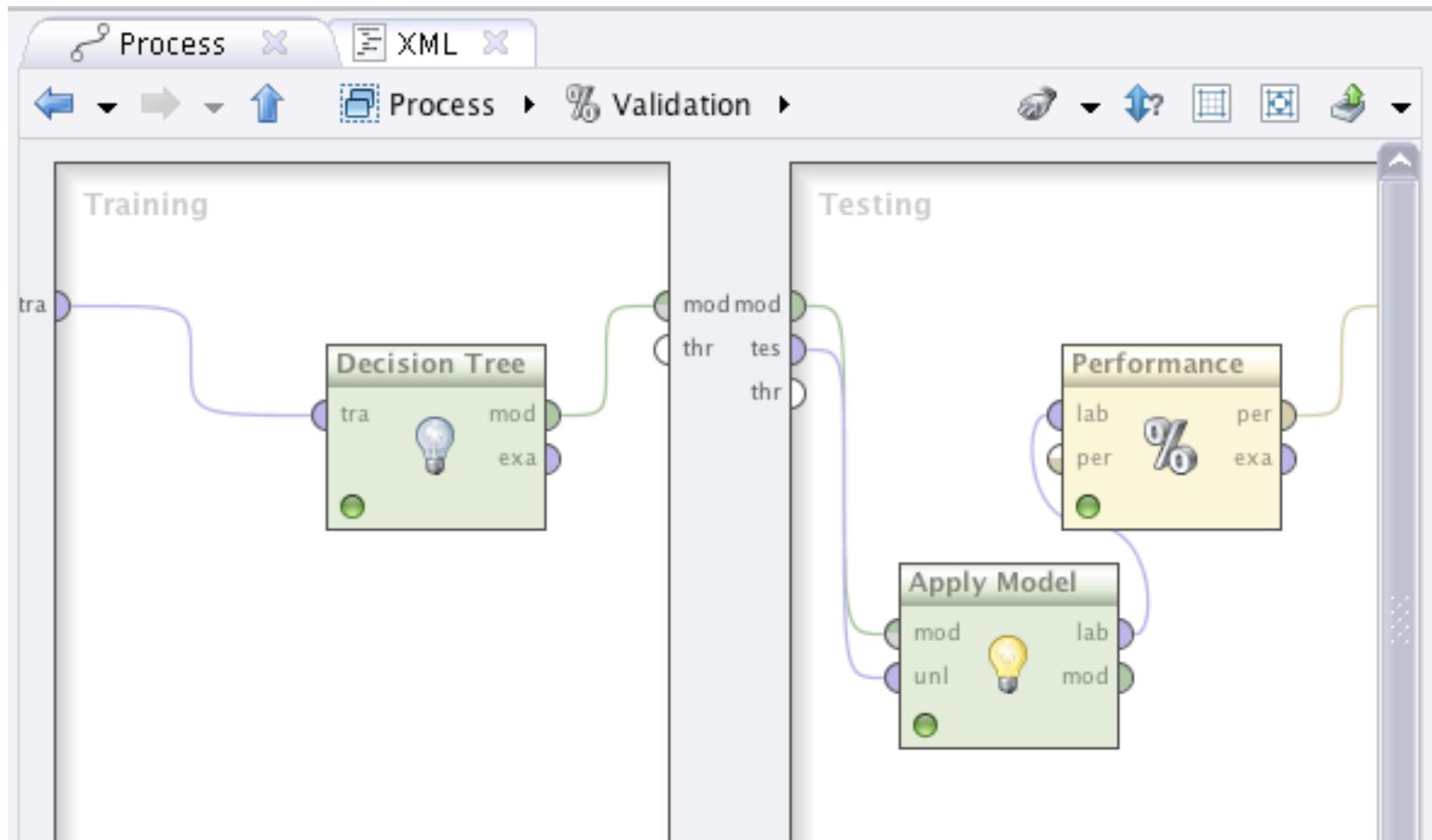
Selected Feature Set



<https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96>

- advanced issues in data preparation
 - data reduction
- ... and modeling
 - dealing with unbalanced classes
 - context
 - resampling
 - create synthetic data with SMOTE
 - cost sensitive learning
- final reflections on data quality

you developed a model for the competition...



accuracy = 99.9%!!



... wth?!

 InClass Prediction Competition

To loan or not to loan - that is the question

Practical assignment of the Machine Learning course at U.Porto

58 teams · 19 days to go

Overview Data Code Discussion **Leaderboard** Rules Team Host My Submissions **Submit Predictions** ...

[Public Leaderboard](#) [Private Leaderboard](#)

This leaderboard is calculated with approximately 50% of the test data. [Raw Data](#) [Refresh](#)

The final results will be based on the other 50%, so the final standings may be different.

#	Team Name	Notebook	Team Members	Score	Entries	Last
58	G45 - auntdulce		  	0.47160	2	17h

all predictions = negative class!

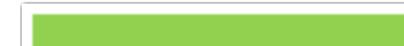


**maybe I should have done the
exploratory data analysis**

(as advised by the lecturers...)



negatives



positives

so what?

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

- ML methods usually minimize FP+FN
 - ... or, at the very least give the same weight to both types of errors
- ... but potentially FP >> FN
 - i.e. quality of the model more affected by FP
- ... so algorithm effectively minimizes FP!
- ... and there's an easy model for that
 - prediction = N

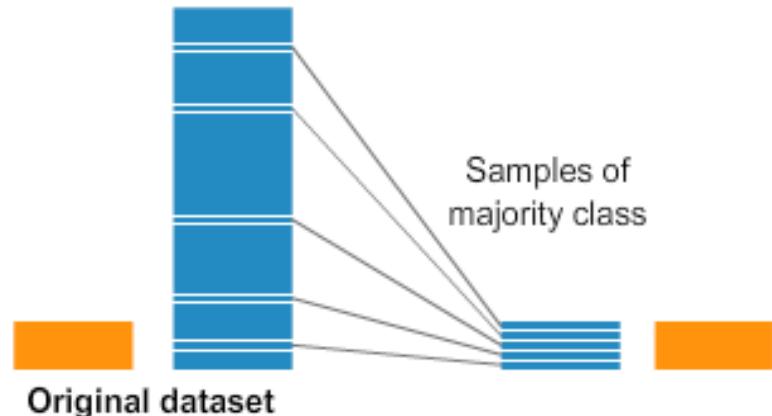
- advanced issues in data preparation
 - data reduction
- ... and modeling
 - dealing with unbalanced classes
 - context
 - resampling
 - create synthetic data with SMOTE
 - cost sensitive learning
- final reflections on data quality

learning with class imbalance

- collect more data
 - difficult in many domains
- resample existing data
 - delete data from the majority class
 - duplicate data from the minority class
- create synthetic data
 - e.g. SMOTE
- adapt your learning algorithm
 - e.g. cost sensitive learning

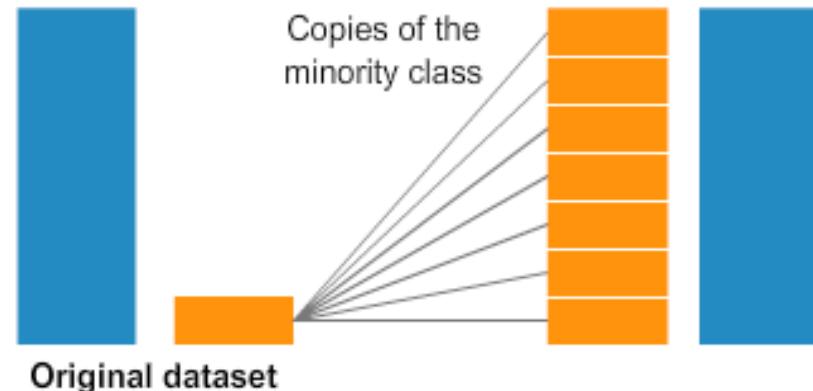
resampling

Undersampling



<https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets#t1>

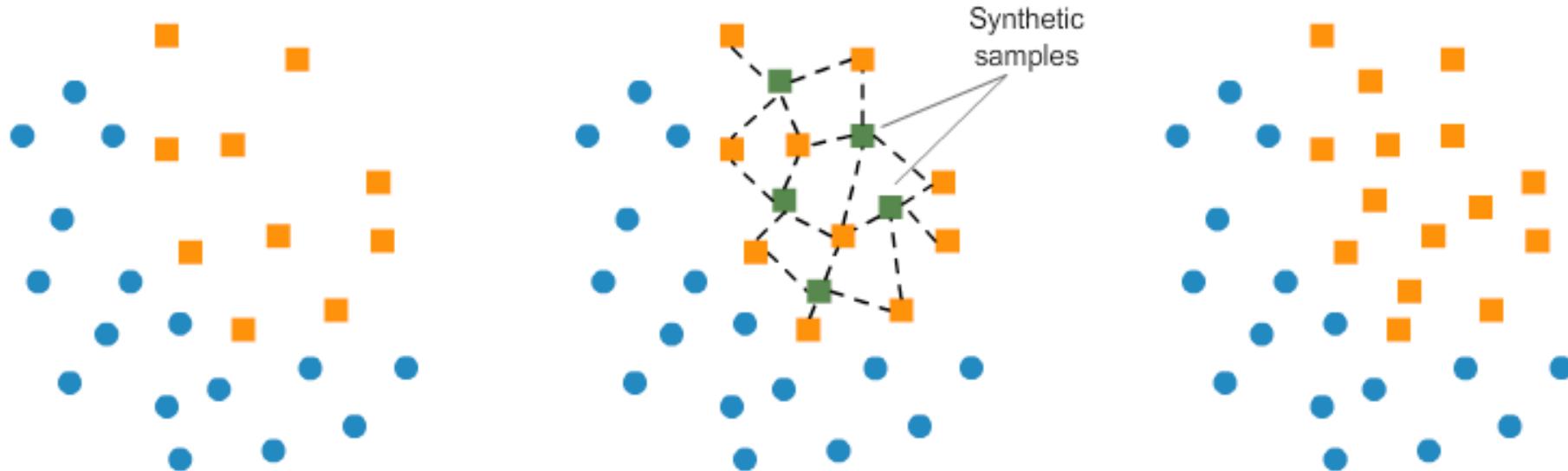
Oversampling



- ... but possible loss of information
- ... but fixed boundaries
- ... and danger of overfitting

furthermore, what is the best ratio?

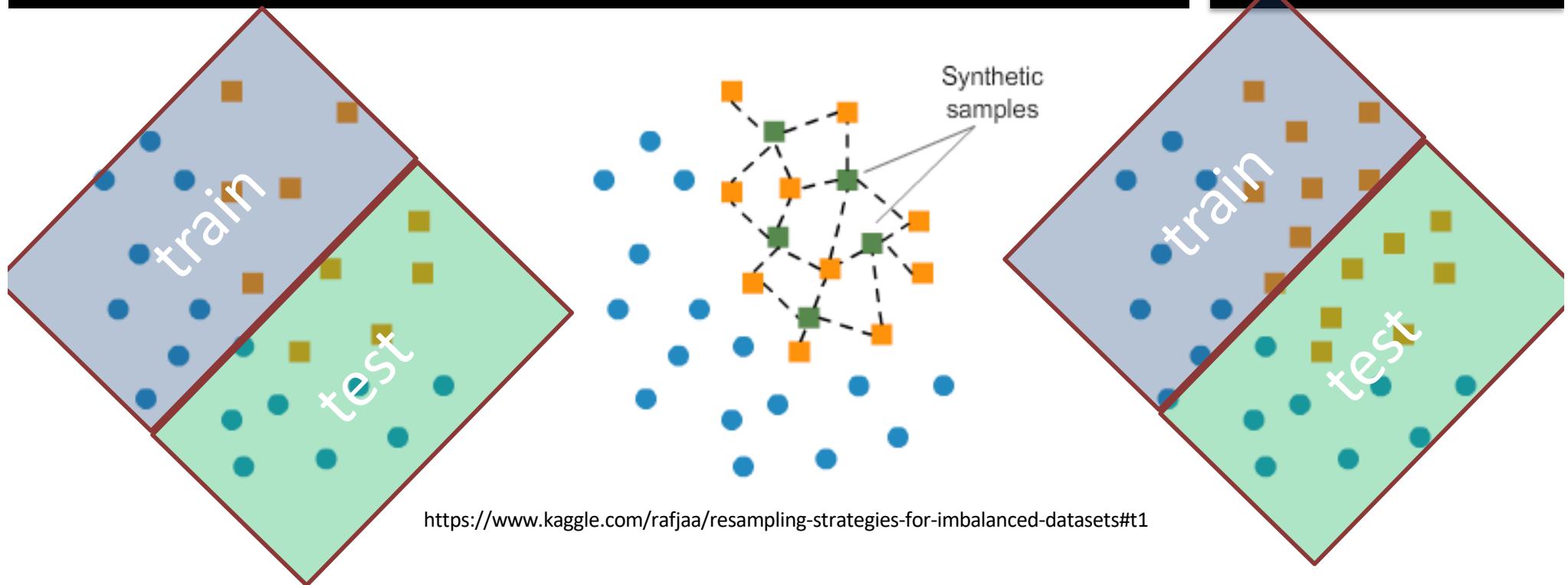
SMOTE (Synthetic Minority Over-sampling Technique)



<https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets#t1>

- ... but possibility of inadequate boundaries
 - what happens if minority observations are too far apart?
- ... and danger of overfitting

SMOTE + lack of basic statistical knowledge?



which one?
(or both?)

- advanced issues in data preparation
 - data reduction
- ... and modeling
 - dealing with unbalanced classes
 - context
 - resampling
 - create synthetic data with SMOTE
 - cost sensitive learning
- final reflections on data quality

the cost of errors

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

- FP and FN errors often incur different costs
 - medical diagnostic
 - loan decisions
 - marketing campaigns
 - fraud detection in bank transactions
 - fault detection in machines
- ... but ML methods still usually minimize FP+FN

error vs missclassification costs: medical diagnosis example

unnecessary suffering + more expensive
procedures + eventually death
e.g. 100

		Predicted class	
		Yes	No
Actual class	Yes	10	5
	No	5	80

unnecessary exams & anxiety
e.g. 5

- error = $10 / 100 = 10\%$
- missclassification costs
 $= 5 \times 100 + 5 \times 5 = 525$
- ... per patient
– 5.25

cost-sensitive learning

- simple methods
 - resampling according to costs
 - weighting according to costs
 - basically, the same thing
- complex methods
 - e.g. metacost

1. create bootstrap replicates of training data
2. learn model from each replicate
3. relabel examples

$$\operatorname{argmin}_i \sum_j P(j|x)C(i,j)$$

- $C(i | j)$ = cost of mistaking j by i
 - $P(j | x)$ = class probability of x by voting
4. learn model on relabelled data
-
- independent of algorithm

- advanced issues in data preparation
 - data reduction
- ... and modeling
 - dealing with unbalanced classes
- final reflections on data quality

coming next...

- data
- cleaning
 - data quality: a data scientist's worst nightmare
 - quality issues
 - can we do better?
- integration
- reduction
- transformation and discretization
- challenges

data quality: multidimensional view

- accuracy
 - correct or wrong, accurate or not
- completeness
 - not recorded, unavailable, ...
- consistency
 - some modified but some not, dangling, ...
- timeliness
 - timely update?
- believability
 - how trustable is the data and its sources?
- interpretability
 - how easily the data can be understood?

a data scientist's worst nightmare

no worries: our data is clean!



no worries: our data is clean!

(1/6)

we have a data warehouse

- DWs are built with a different purpose
 - descriptive
 - aggregated data
 - typically



<https://pixabay.com/en/archive-bookcase-boxes-business-1850170/>

no worries: our data is clean! (2/6)

our IS was just revamped

- how far was analytics taken into account in the process?



<https://pixabay.com/en/confused-muddled-illogical-880735/>

no worries: our data is clean! (3/6)

we had a major data cleanup

- how was project success measured?



<https://pixabay.com/en/clean-dare-cleaning-1706439/>

no worries: our data is clean! (4/6)

our data is collected automatically

- ... and we all know that machines never break

A problem has been detected and windows has been shut down to prevent damage to your computer.

The problem seems to be caused by the following file: SPCMDCON.SYS

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)

*** SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, Datestamp 3d6dd67c

https://en.wikipedia.org/wiki/Blue_Screen_of_Death#/media/File:Windows_XP_BSOD.png

no worries: our data is clean! (5/6)

our data collection is human-error proof

- “Data errors, uh, find a way”



<http://knowyourmeme.com/memes/life-uh-finds-a-way>

no worries: our data is clean! (6/6)

tell us what you need: we have everything

- should be read as

“if something goes wrong,
it’s your fault”
- ... often associated with

“you do magic, right?”



[https://en.wikipedia.org/wiki/Marvin_\(character\)#/media/File:Marvin_\(HHGG\).jpg](https://en.wikipedia.org/wiki/Marvin_(character)#/media/File:Marvin_(HHGG).jpg)

human resources

- remember when IT director was not a C-level job?



Fair use, <https://en.wikipedia.org/w/index.php?curid=24782741>

can we do any better? (2/3)

analytics at the core of IS development

- requirements analysis includes analytics
- analytics components built in the same process as the rest of the functionalities
- we don't mind that systems still do old-fashioned tasks
 - sell stuff, pay salaries, etc.



<https://pixabay.com/en/scaffolding-workers-construction-1617969/>

can we do any better? (3/3)

data quality is a continuous process

- data steward
- ... the sexiest job of the XXII century?



<https://pixabay.com/en/lost-places-toilet-urinal-pforphoto-1610652/>

data cleaning as a process

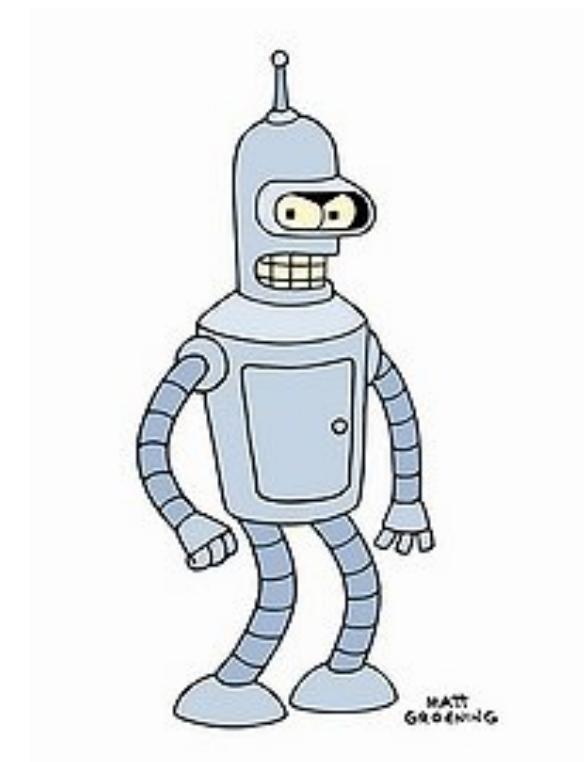
- discrepancy detection
 - validate with metadata (e.g., domain, range, dependency, distribution)
 - check field overloading
 - check uniqueness rule, consecutive rule and null rule
 - commercial tools
 - scrubbing: use simple domain knowledge to detect errors and make corrections
 - e.g. postal code, spell-check
 - auditing: discover rules and relationship to detect violators
 - e.g. correlation and clustering to find outliers
- migration and integration
 - data migration tools
 - allow transformations to be specified
 - ETL (Extraction/Transformation>Loading) tools
 - allow users to specify transformations through a graphical user interface

Summary

- data
- ... quality
 - accuracy, completeness, consistency, timeliness, believability, interpretability
- ... cleaning
 - e.g. missing/noisy values, outliers
- integration from multiple sources
 - entity identification problem is challenging
- reduction
 - curse of dimensionality and dimensionality reduction
 - numerosity reduction
- transformation and discretization

automation

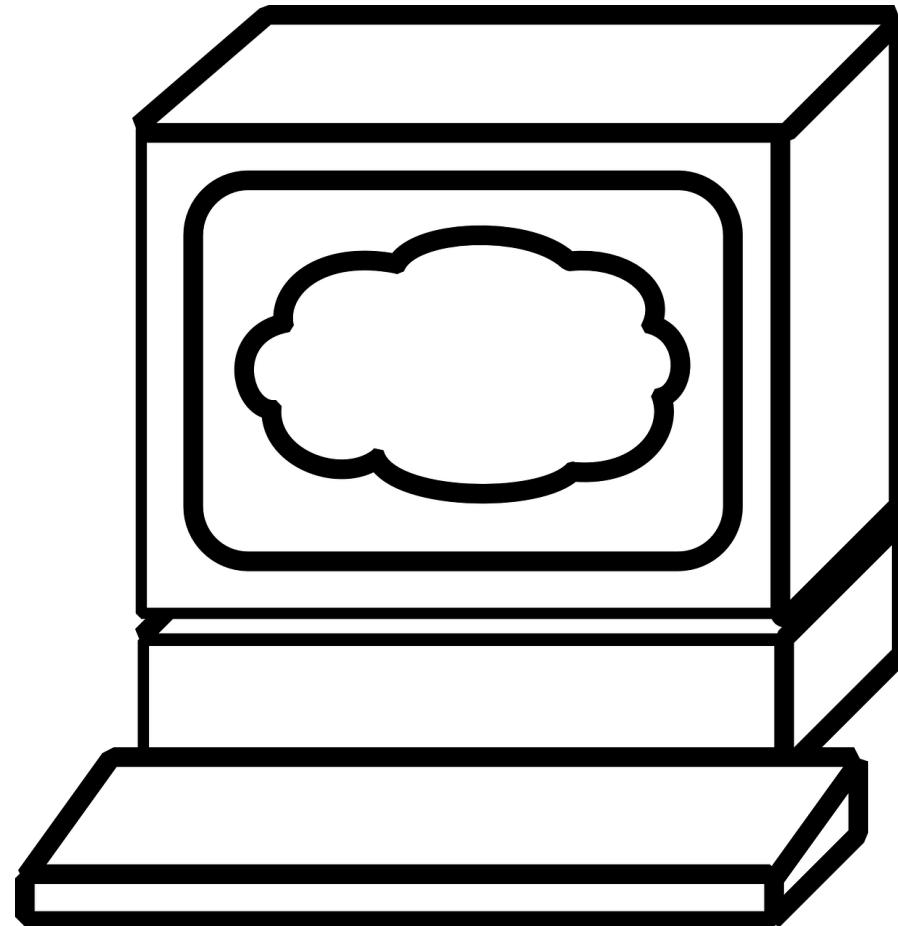
- automl & metalearning
 - some progress on algorithm selection
 - ... early work on workflow
 - including data preparation
 - ... not really data cleaning



[https://pt.wikipedia.org/wiki/Ficheiro:Bender_\(personagem\).jpg](https://pt.wikipedia.org/wiki/Ficheiro:Bender_(personagem).jpg)

DQaaS?

- if automation is possible
 - DQ can become a commodity?
- perhaps there is hope?
- ... many issues
 - confidentiality
 - computational costs



<https://pixabay.com/en/computer-server-internet-network-294036/>

Descriptive Modelling

Rita P. Ribeiro

Machine Learning - 2021/2022



DEPARTAMENTO DE CIÉNCIA DE COMPUTADORES
FACULDADE DE CIÉNCIAS DA UNIVERSIDADE DO PORTO

Summary

- Descriptive Analytics
- Clustering Methods

Descriptive Analytics

Descriptive Analytics

Goals:

- Describe/summarize or finding structure on what we have observed
 - Data summarization and visualization (e.g. PCA) can be seen as simple forms of descriptive analytics
 - However, most frequently descriptive modeling is associated with clustering

Similarity Measures

- How to measure similarity between objects?
- The notion of similarity is strongly related with the notion of distance between observations
- It can be measured as the oposite of the distance

ID	Income	Position	Age
1	2500	manager	35
2	2750	manager	30
3	4550	director	50

- Which cases are more similar?

Similarity Measures (cont.)

- Similarity measure
 - Numerical measure of how alike two data objects are.
 - Is higher when objects are more alike.
 - Often falls in the range [0,1]
- Dissimilarity measure
 - Numerical measure of how different two data objects are
 - Lower when objects are more alike
 - Minimum dissimilarity is often 0 Upper limit varies

Proximity refers to a similarity or dissimilarity

Similarity Measures (cont.)

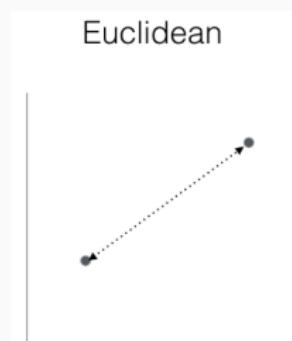
- Dissimilarity measure can be expressed by a distance metric
- Distance metrics d have some well-known properties
 - Given any two data points x_i and x_j
 - $d(x_i, x_j) \geq 0$
 - $d(x_i, x_j) = 0$ only if $x_i = x_j$
 - $d(x_i, x_j) = d(x_j, x_i)$
 - $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$ for any point x_i , x_j and x_k - triangle inequality

Similarity Measures (cont.)

Euclidean Distance

$$d(x_i, x_j) = \sqrt{\sum_{a=1}^n (x_i^a - x_j^a)^2}$$

where n is the number of attributes and x_i^a and x_j^a are the a^{th} attribute value for the data points x_i and x_j , respectively



Similarity Measures (cont.)

Manhattan Distance

$$d(x_i, x_j) = \sum_{a=1}^n |x_i^a - x_j^a|$$

where n is the number of attributes and x_i^a and x_j^a are the a^{th} attribute value for the data points x_i and x_j , respectively



Similarity Measures (cont.)

A Generalization: **Minkowski Distance**

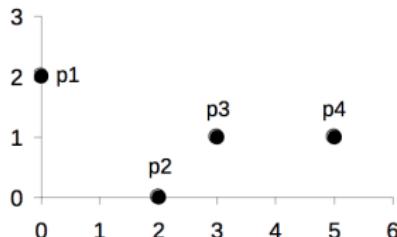
$$d(x_i, x_j) = \sqrt[p]{\sum_{a=1}^n |x_i^a - x_j^a|^p}$$

where if

- $p = 1$, we have the Manhattan Distance (or L_1 -norm)
- $p = 2$, we have the Euclidean Distance (or L_2 -norm)
- ...
- $p = \infty$, we have Chebyschev or *supremum* distance (or L_∞ -norm): it gives the maximum difference between any of the attributes of the data points.

Similarity Measures (cont.)

Example of Minkowski Distances: L_1 -norm, L_2 -norm and L_∞ -norm



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Similarity Measures (cont.)

- More examples of similarity/distance measures
 - Canberra distance
 - Jaccard Coefficients
 - Cosine similarity
- Still, several problems may arise that may distort the notion of distance:
 - different scales of variables
 - different importance of variables
 - different types of data (e.g. both numeric and categorical variables)

Similarity Measures (cont.)

Heterogeneous Distance Functions

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{a=1}^n \delta_a(x_i^a, x_j^a)$$

where

- if a is a categorical variable

$$\delta_a(x_i^a, x_j^a) = \begin{cases} 0 & \text{if } x_i^a == x_j^a \\ 1 & \text{otherwise} \end{cases}$$

- if a is a numeric variable

$$\delta_a(x_i^a, x_j^a) = \frac{|x_i^a - x_j^a|}{|max_a - min_a|}$$

Similarity Measures (cont.)

General Coefficient of Similarity

$$s(x_i, x_j) = \sum_{a=1}^n w_a s(x_i^a, x_j^a) / \sum_{a=1}^n w_a$$

$s()$ is a similarity measure, n is the number of attributes, x_i^a and x_j^a are the a^{th} attribute value for the data points x_i and x_j , respectively, and w_a is a value between 0 and 1 corresponding to the weight contribution of the attribute a .

- Similarity measures, also have some well known properties
 - Given any two data points x_i and x_j
 - $s(x_i, x_j) = 1$, only if $x_i = x_j$
 - $s(x_i, x_j) = s(x_j, x_i)$

Clustering

Clustering

Goals:

- Obtain the “natural” grouping of a set of data - i.e. find some structure on the data set
 - The key issue on clustering is the notion of similarity
 - Observations on the same group are supposed to share some properties, i.e. being similar
 - Most methods use the information on the distances among observations in a data set to decide on the natural groupings of the cases
- Provide some abstraction of the found groups (e.g. a representation of their main features; a prototype for each group; etc.), gain novel insights of data

Clustering: Some Applications

- Biology
 - describe spatial and temporal communities of organisms
 - group genes or proteins that have similar functionality
- Business and Marketing
 - describe different market segments from a set of potential clients
 - group stocks with similar price fluctuations
- Web Mining
 - find groups of related documents for information retrieval
 - find communities in social networks
 - build recommender systems
- ...

Clustering: Main Types of Methods

- **Partitional**: divide the observations in k partitions according to some criterion
- **Hierarchical**: generate a hierarchy of groups, from 1 to n groups, where n is the number of lines in the data set
 - **Agglomerative**: generate a hierarchy from bottom to top (from n to 1 group)
 - **Divisive**: create a hierarchy in a top down way (from 1 to n groups)

Clustering Partitional Methods

Goal: Partition the given set of data into k groups by either minimizing or maximizing a pre-specified criterion

- Some key issues:
 - The user needs to select the number of groups
 - The number of possible divisions of n cases into k groups can grow fast!

$$N(n, k) = \frac{1}{k!} \sum_{i=1}^k (-1)^{k-i} \binom{k}{i} i^n$$

e.g. for $n = 100$ and $k = 5$, $N(100, 5) \approx 6.6 \cdot 10^{67}$

Clustering Partitional Methods (cont.)

Some important properties

- Cluster compactness
 - how similar are cases within the same cluster
- Cluster separation
 - how far is the cluster from the other clusters
- The goal is to **minimize intra-cluster distance** and **maximize inter-cluster distances**.
- A clustering solution assigns all the objects to a cluster
 - *hard clustering*: an object belongs to a single cluster
 - *fuzzy clustering*: each object has a probability associated to belong to each cluster

Clustering Partitional Methods (cont.)

Consider the cluster $C_k = \{x_1, x_2, \dots, x_{n_k}\}$, the **centroid** of C_k is given by

$$\bar{x}^{(k)} = \frac{1}{n_k} \sum_{x_i \in C_k} x_i$$

the centroid of C_k can also be the median of its data objects, i.e. $\tilde{x}^{(k)}$

Goal: obtain a **set of clusters C** that minimize

$$h(C) = \sum_{j=1}^k \sum_{x_i \in C_j} d(x_i, \bar{x}^{(j)})$$

(Some) Criteria for numeric data

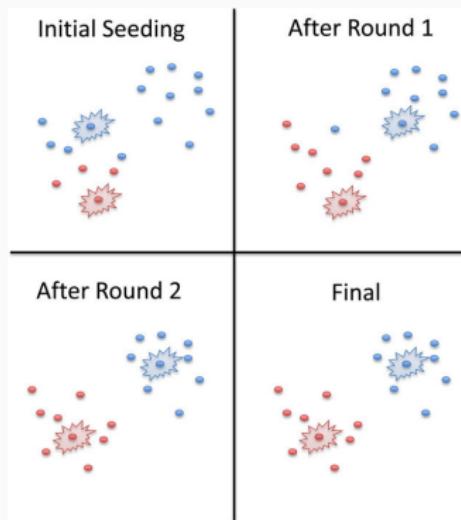
- Sum of Squared Errors (SSE): $d(x_i, \bar{x}^{(j)}) = (x_i - \bar{x}^{(j)})^2$
- L_1 measure: $d(x_i, \bar{x}^{(j)}) = |x_i - \bar{x}^{(j)}|$

Clustering Partitional Methods: k -Means

It is a partition-based method that obtains k groups of a data set

k -means algorithm

- Initialize the centers of the k groups to a set of randomly chosen observations
- Repeat
 - Allocate each observation to the group whose center is nearest
 - Re-calculate the center of each group
- Until the groups are stable, i.e. there is no significant decrease or there is an increase on the minimize criterion $h(C)$



Clustering Partitional Methods: k -Means (cont.)

Some observations:

- It uses the squared Euclidean distance as criterion
- Maximizes inter-cluster dissimilarity

Advantages:

- Fast algorithm that scales well
- Stochastic approach that frequently works well. It tends to identify local minima.

Disadvantages:

- It does not ensure an optimal clustering
- We may obtain different solutions with different starting points
- The initial guess of k for the number of clusters, maybe away from the real optimal value of k .

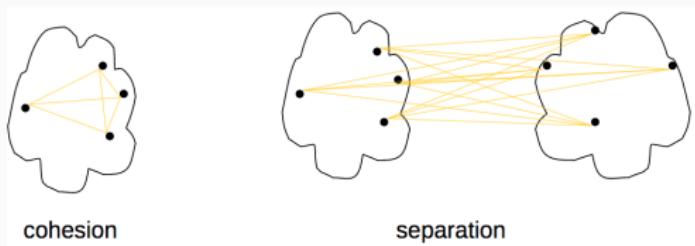
Clustering Validation

How to validate/evaluate/compare the results obtained by some clustering method?

- Is the found group structure random?
- What is the “correct” number of groups?
- How to evaluate the result of a clustering algorithm when we do not have information on the number of groups in the data set?
- How to compare the results obtained by different methods when outside information on the number of groups exists?
- How to compare alternative solutions (e.g. obtained using different clustering algorithms)?

Clustering Validation: Types of Evaluation Measures

- **Supervised** - compare the obtained clustering (grouping) with the external information that we have available
- **Unsupervised** - try to measure the quality of the clustering without any information on the “ideal” structure of the data
 - Cohesion coefficients - determine how compact/cohesive are the members of a group
 - Separation coefficients - determine how different are the members of different groups



Clustering Validation: Silhouette Coefficient

Silhouette Coefficient (unsupervised measure)

- Popular coefficient that incorporates both the notions of cohesion and separation
- For each object x_i :
 - obtain the average distance to all objects in the same group (a_i)
 - to any other group to which x_i does not belong, calculate the average distance to the members of these other groups; obtain the minimum value of these distances (b_i)
 - The silhouette coefficient, s_i is equal to

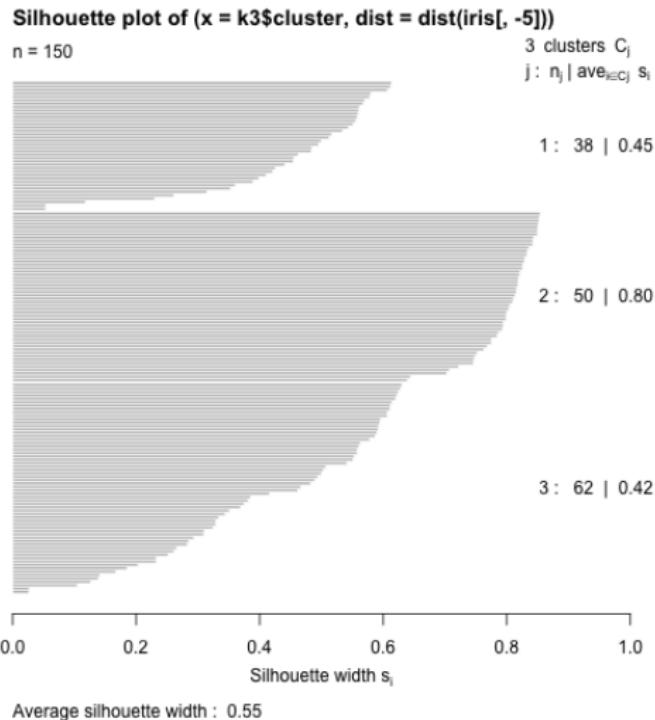
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- The coefficient takes values between -1 and 1 .

Clustering Validation: Silhouette Coefficient (cont.)

Example: iris data set silhouette coefficients s_i with $k = 3$ clusters

- Large s_i (almost 1) means that they are very well clustered.
- Small s_i (around 0) means that they lie between two clusters.
- Negative s_i means that they are probably placed in the wrong cluster.
- The closer average silhouette to 1, the better.



Clustering Validation: Best Number of Clusters

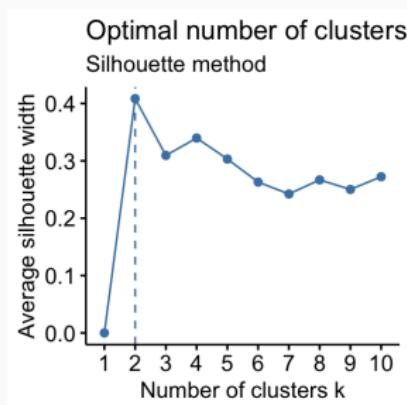
How to select the right k for k-means?

- An inappropriate choice of k can result in a clustering with poor performance.
- What happens if we select a k that is too high? What if the k is too low?
- Ideally, you should have some a priori knowledge on the real structure of the data.
- If no a priori value is known start with $\sqrt{n/2}$ as a rule of thumb, where n is the number of attributes.

Clustering Validation: Best Number of Clusters (cont.)

For several possible number of clusters k :

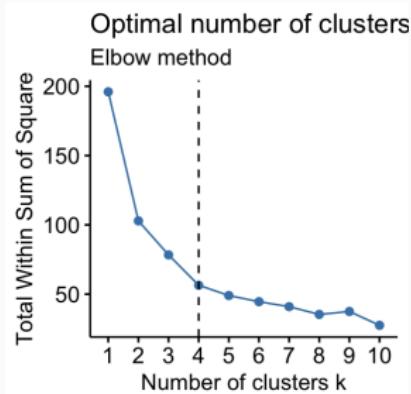
- Calculate the average silhouette coefficient value and choose the k that yields to the highest value



Clustering Validation: Best Number of Clusters (cont.)

(Elbow method) For several possible number of clusters k :

- Calculate the within-cluster SSE, also called distortion, and choose the k so that adding another cluster doesn't yield to a much smaller SSE.



Other, more sophisticated methods exist (e.g. intracluster to intercluster distance ratio)

PAM (Partitioning Around Medoids)

- It searches for the k representative objects (the medoids) among the cases in the given data set.
- As with k-means each observation is allocated to the nearest medoid.
- Is more robust to the presence of outliers because it uses original objects as centroids instead of averages that may be subject to the effects of outliers.
- Moreover, it uses a more robust measure of the clustering quality: $L_1 - norm$, which is based on absolute error instead of the squared error used in k-means,

CLARA (Clustering Large Applications)

- The PAM algorithm has several advantages in terms of robustness when compared to k-means.
- However, these advantages come at the price of additional computational complexity that may be too much for very large data sets
- CLARA tries to solve these efficiency problems
 - It does that by using sampling, i.e. working on parts of the data set instead of the full data set

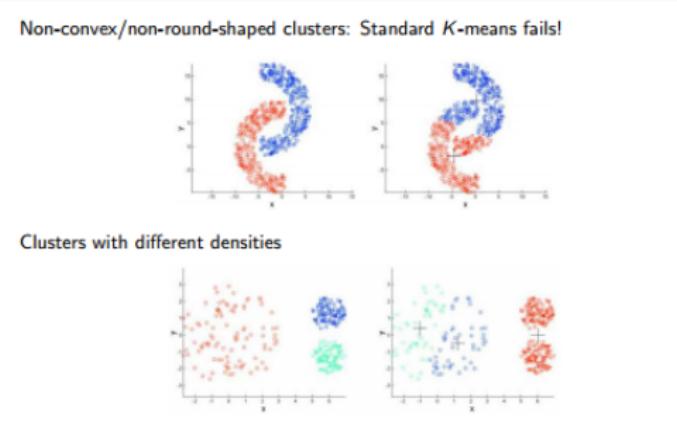
CLARA Algorithm

- Repeat n times the following:
 - Draw a random sample of size m
 - Apply PAM to this random sample to obtain k centroids
 - Allocate the full set of observations to one of these centroids
 - Calculate sum of dissimilarities of the resulting clustering (as in PAM)
- Return as result the clustering of the n repetitions that got lowest sum of dissimilarities

Other Clustering Partitional Methods (cont.)

These “k-means like” methods have problems when:

- clusters are of different sizes, densities and with non-globular shape



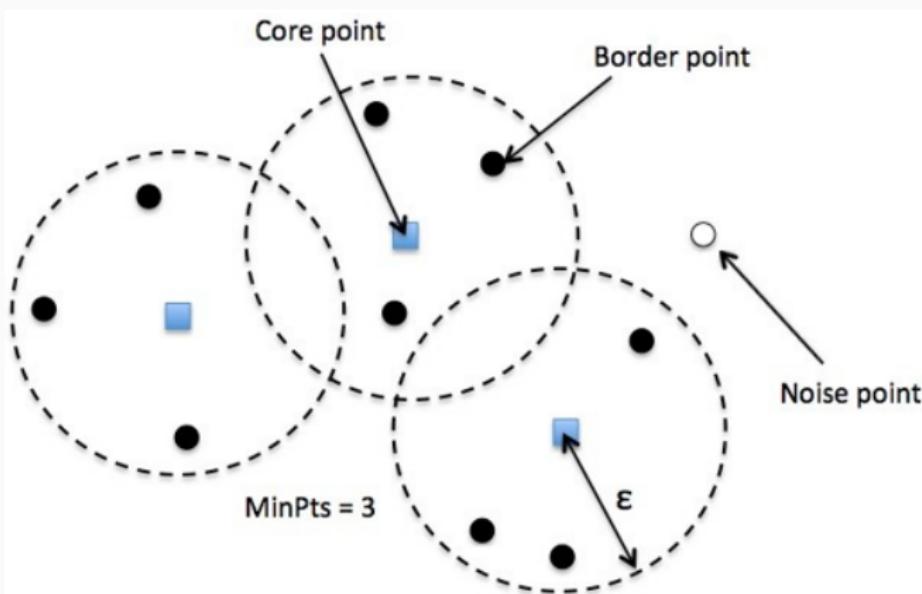
- data contains outliers/noise

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- The density of a single observation is estimated by the number of observations that are within a certain radius (a parameter of the method)
- Based on this idea observations are classified as:
 - core points: if the number of observations within its radius are above a certain threshold
 - border points: if the number of observations within their radius does not reach the threshold but they are within the radius of a core point
 - noise points: they do not have enough observations within their radius, nor are they sufficiently close to any core point

Other Clustering Partitional Methods (cont.)

DBSCAN: Core, Border and Noise Points



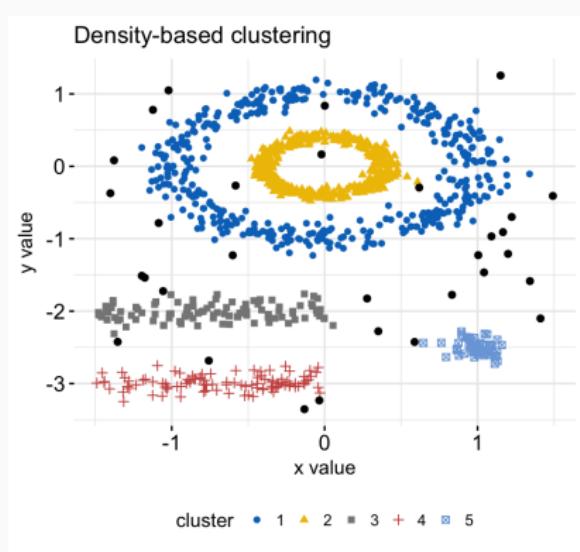
Other Clustering Partitional Methods (cont.)

- DBSCAN Algorithm
 - Classify each observation in one of the three possible alternatives
 - Eliminate the noise points from the formation of the groups
 - All core points that are within a certain distance of each other are allocated to the same group
 - Each border point is allocated to the group of the nearest core point
- Note that this method does not require the user to specify the number of groups.
- But, you need to specify the radius (ε) and the minimum number of points (MinPts)

Other Clustering Partitional Methods (cont.)

DBSCAN

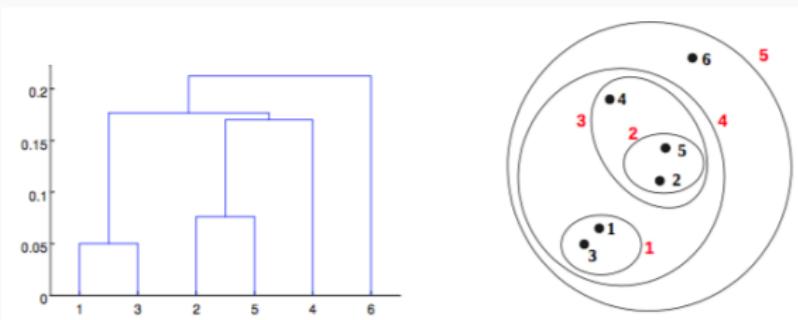
- Advantages:
 - Can handle clusters with different shapes and sizes
 - Resistant to noise
- Disadvantages:
 - Varying densities
 - High-dimensional data



Hierarchical Clustering

Goal:

- Obtain a hierarchy of groups, where each level represents a possible solution with x groups. It is up to the user to select the solution he wants.
- A dendrogram can be used for visualization



Hierarchical Clustering (cont.)

- **Agglomerative Methods** - *bottom-up*
 - Start with as many groups as there are cases
 - On each upper level a pair of groups is merged into a single group
 - The chosen pair is formed by the groups that are more similar
- **Divisive Methods** - *top-down* (much less used)
 - Start with a single group
 - On each level select a group to be split in two
 - The selected group is the one with smallest uniformity

Hierarchical Clustering (cont.)

Some proximity measures for the merging/splitting step

single link

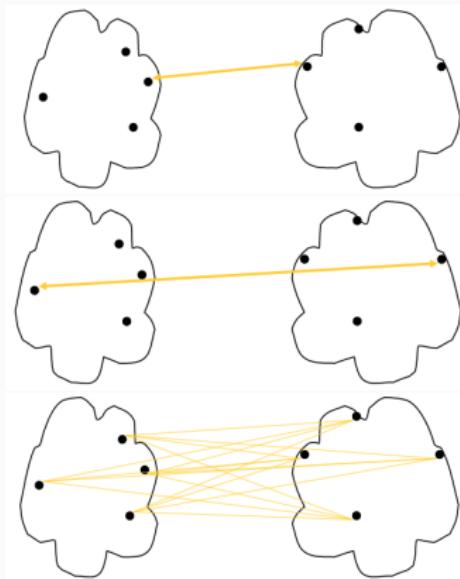
$$d(C_1, C_2) = \min_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$

complete link

$$d(C_1, C_2) = \max_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$

average link

$$d(C_1, C_2) = \frac{1}{n_1 n_2} \sum_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$



Other methods also exist (e.g. distance between the centroids, Ward's method that uses SSE).

Hierarchical Clustering: Agglomerative Methods

Algorithm

- Compute the proximity matrix
- Let each data point be a cluster
- Repeat
 - Merge the two closest clusters
 - Update the proximity matrix
- Until only a single cluster remains

Hierarchical Clustering: Agglomerative Methods (cont.)

Example: Consider the following distance matrix

	A	B	C	D	E	F
A	0					
B	4	0				
C	25	21	0			
D	24	20	1	0		
E	9	5	16	15	0	
F	7	3	18	17	2	0

Distance Matrix - Stage 0

Use Agglomerative Hierarchical Clustering to obtain the single-link dendogram.

Hierarchical Clustering: Agglomerative Methods (cont.)

Example: Agglomerative Hierarchical Clustering, single-link method.

	A	B	C	D	E	F
A	0					
B	4	0				
C	25	21	0			
D	24	20	1	0		
E	9	5	16	15	0	
F	7	3	18	17	2	0

Distance Matrix - Stage 0

	A	B	CD	E	F
A	0				
B	4	0			
CD	24	20	0		
E	9	5	15	0	
F	7	3	17	2	0

Distance Matrix - Stage 1

	A	B	CD	EF
A	0			
B	4	0		
CD	24	20	0	
EF	7	3	15	0

Distance Matrix - Stage 2

	A	BEF	CD
A	0		
BEF	4	0	
CD	24	15	0

Distance Matrix - Stage 3

	ABEF	CD
ABEF	0	
CD	15	0

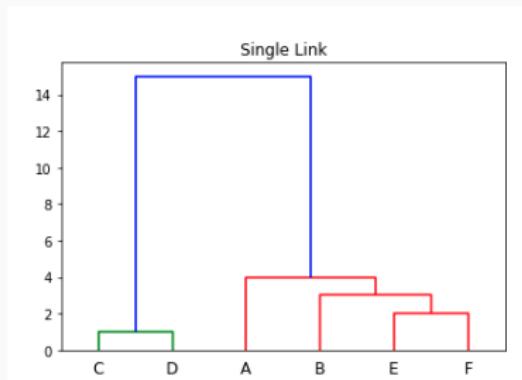
Distance Matrix - Stage 4

Hierarchical Clustering: Agglomerative Methods (cont.)

Example: Agglomerative Hierarchical Clustering, single-link method.

	A	B	C	D	E	F
A	0					
B	4	0				
C	25	21	0			
D	24	20	1	0		
E	9	5	16	15	0	
F	7	3	18	17	2	0

Distance Matrix - Stage 0



Hierarchical Clustering: Agglomerative Methods (cont.)

Different proximity measures yield to different types of clusters.

- single-link
 - can handle non-elliptical shapes
 - uses a local merge criterion
 - distant parts of the cluster and the clusters' overall structure are not taken into account

Hierarchical Clustering: Agglomerative Methods (cont.)

Different proximity measures yield to different types of clusters.

- complete-link
 - biased towards globular clusters
 - uses a non-local merge criterion
 - chooses the pair of clusters whose merge has the smallest diameter
 - the similarity of two clusters is the similarity of their most dissimilar members
 - sensitive to noise/outliers
- average-link
 - it is a compromise between single and complete link

Hierarchical Clustering: Divisive Methods

Algorithm

- Compute the proximity matrix
- Start with a single cluster that contains all data points
- Repeat
 - choose the cluster with the largest diameter, i.e. largest dissimilarity between any two of its points
 - select the data point with largest average dissimilarity to the other members in that cluster
 - re-allocate the data points to either the cluster of this selected point or the “old” cluster (represented by its center), depending on which one is nearest
- Until each data point constitutes a cluster

Clustering Methods: Wrap-up

Overall, we can compare clustering methods w.r.t

- Algorithm:
 - complexity and scalability
 - similarity measures that can be employed
 - robustness to noise
 - it is able to find clusters on sub-spaces
 - different runs lead to different results
 - it is incremental

Clustering Methods: Wrap-up (cont.)

- Data:
 - it is able to handle different types of data (continuous, categorical, binary)?
 - is there dependency on the order of data points?
- Domain:
 - does the algorithm finds the number of clusters, or needs it as input?
 - how many parameters are necessary?
 - what is the required domain knowledge for that?
- Results:
 - shape of clusters that is able to find
 - interpretability

References

References

- Aggarwal, Charu C. 2015. *Data Mining, the Texbook*. Springer.
- Gama, João, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. 2015. *Extração de Conhecimento de Dados: Data Mining -3rd Edition*. Edições Sílabo.
- Gandomi, Amir, and Murtaza Haider. 2015. "Beyond the Hype: Big Data Concepts, Methods, and Analytics." *International Journal of Information Management* 35 (2): 137–44. doi:<https://doi.org/10.1016/j.ijinfomgt.2014.10.007>.
- Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- "R Project." 2021. <https://www.r-project.org/>.
- Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining*. 2nd ed. Pearson.

Association Rules

Rita P. Ribeiro

Machine Learning - 2021/2022



Summary

1. Association Rules in Action
2. Association Rules Basic Concepts
3. Mining Association Rules
 - Problem Definition
 - Apriori Algorithm
 - Compact Representation of Itemsets
 - Selection of Rules
 - Apriori variants: FP-growth
 - Conclusions

Association Rules in Action

Association Rules: a New Data Mining Task

Data Mining Tasks:

- Predictive
 - Classification
 - Regression
 - ...
- Descriptive
 - Clustering
 - **Association Rules**
 - find relationships / associations between groups of variables
 - ...

Motivation

Originally stated in the context of Market Basket Analysis

- Data consists of set of items bought by customers, referred as **transactions**
- Find unexpected associations between sets of items using the frequency of sets of items
- Discovered sets of items are referred as **frequent itemsets** or **frequent patterns**
- Goals
 - Store layout - *Should products A and B be placed together?*
 - Promotions - *If the client is interested in {A,B,C,...}, can we guess other interests?*
 - ...

Actionable Knowledge: shop layout

- Possible actions from rule $\{A1, A4\} \rightarrow \{A6\}$
 - Sell the $A1, A4, A6$ together (pack)
 - Place article $A6$ next to articles $A1, A4$
 - Offer a discount coupon for $A6$ in articles $A1, A4$
 - Place a competitor of $A6$ next to $A1, A4$ (brand protection).
- Note
 - These actions must make sense from the business point of view.



Actionable Knowledge: cross selling

- Steps
 - Client puts article *A* in basket
 - Shop knows rule $A \rightarrow B$
 - Rule has enough confidence ($> 20\%$)
 - Shop tells client he may be interested in *B*
 - Client decides whether to buy *B* or not
- Notes
 - Rules are discovered from business records
 - Discovery (mining) can be made off-line
 - Use of rules can be made on-line



Actionable Knowledge: text mining

- Each document is treated as a “bag” of terms and keywords
 - doc1: Student, Teach, School (Education)
 - doc2: Student, School (Education)
 - doc3: Teach, School, City, Game (Education)
 - doc4: Baseball, Basketball (Sport)
 - doc5: Basketball, Player, Spectator (Sport)
 - doc6: Baseball, Coach, Game, Team (Sport)
 - doc7: Basketball, Team, City, Game (Sport)
- Goal: identify co-occurring terms and keywords
- Example:
 - Student, School → Education
 - Game → Sport

Actionable Knowledge: health

- Rules obtained from the patient's records
- Sooner prevention
- Each patient visits a health unit one or more times
- We record the observations for each visit
 - Symptoms (head ache, temperature)
 - Exam results (blood pressure, sugar level)
- A set of observations may fire a rule
 $\{\text{Head ache, blood pressure rise}\} \rightarrow \{\text{stroke, immobilization}\}$
- When head ache and blood pressure rise are observed, stroke and immobilization are also expected.
- Not necessarily causal

Usage patterns

- Most visited pages
- Frequent page sets
 - Site structure
- Pages associated to users
 - personalization
- Seasonal effects
 - operations, campaigns
- Cross-preferences
 - cross-selling

Association Rules

Basic Concepts

Market Basket Analysis



Market Baskets data set

TID	Products
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

Products are converted in binary flags



TID	A	B	C	D	E
1	1	1	0	0	1
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	0	1	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	0	1	0	0
8	1	1	1	0	1
9	1	1	1	0	0

Market Basket Analysis: how frequent is an itemset?

- Sugar, Flower and Eggs are sold together



- How important is this set?
- **Support** measures the importance of a set
 - Percentage of transactions t containing the set S
 - Absolute support: number of transactions t containing the set S

Market Basket Analysis: how predictive is an itemset?

- Frequent itemsets are used to generate association rules.
- If you buy sugar and flower, you also buy eggs.
- How strong is this rule?
- **Confidence** measures the strength of the rule
 - Percentage of transactions t that having sugar and flower also have eggs



Association Rules: Basic Concepts

- Consider a set of items I
- A transaction t is a subset of items, i.e. $t \subseteq I$
- Given a data set of transactions $D = \{t_i\}_{i=1}^N$
- An **association rule** is defined as an implication $X \rightarrow Y$, where
 - X and Y are itemsets, i.e. $X, Y \subseteq I$
 - $X \neq \emptyset$, $Y \neq \emptyset$ and $X \cap Y = \emptyset$
- $sup(X)$ is the proportion of transactions in D that include the itemset X
- **support**: $sup(X \rightarrow Y) = sup(X \cup Y)$
- **confidence**: $conf(X \rightarrow Y) = sup(X \cup Y)/sup(X)$

Association Rules: an example

Given the data

Transactions ID	Items Bought
100	A, B, C
200	A, C
150	A, D
500	B, E, F



TID	A	B	C	D	E	F
100	1	1	1	0	0	0
200	1	0	1	0	0	0
150	1	0	0	1	0	0
500	0	1	0	0	1	1

- The itemsets with a minimum support of 50%
- Rules with minimum support of 50% and minimum confidence of 50%
 - $A \rightarrow C$
 - $sup(A \rightarrow C) = sup(\{A, C\}) = 50\%$
 - $conf(A \rightarrow C) = sup(\{A, C\}) / sup(\{A\}) = 66.6\%$
 - $C \rightarrow A$
 - $sup(C \rightarrow A) = sup(\{A, C\}) = 50\%$
 - $conf(C \rightarrow A) = sup(\{A, C\}) / sup(\{C\}) = 100\%$

Frequent Itemsets	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

Mining Association Rules

Problem Definition

- Given:
 - data set of transactions D
 - minimal support $minsup$
 - minimal confidence $minconf$
- Obtain:
 - **all** association rules

$$X \rightarrow Y \quad (s = Sup, c = Conf)$$

such that

$$Sup \geq minsup \text{ and } Conf \geq minconf$$

Apriori Algorithm

The **Apriori Algorithm** [Agrawal and Srikant, 1994] works in two steps:

1. Frequent itemset generation

- itemsets with $support \geq minsup$

2. Rule generation

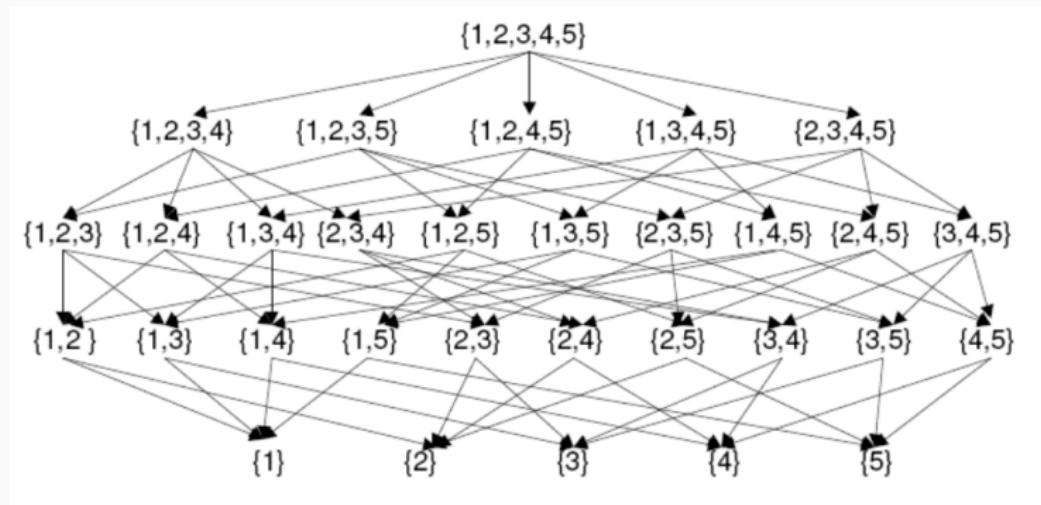
- generate all confident association rules from the frequent itemsets,
i.e. rules with $confidence \geq minconf$

Apriori Algorithm (cont.)

- Problem:
 - there is a very large number of candidate frequent itemsets!
 - for transactions with k items, there are $2^k - 1$ distinct subsets.
- Downward Closure Property
 - every subset of a frequent itemset must also be frequent.
 - ex: if $\{A1, A2, A4\}$ is frequent, so is $\{A1, A2\}$ because every transaction containing $\{A1, A2, A4\}$ also contains $\{A1, A2\}$.
 - thus, every superset of an infrequent itemset is also infrequent.
 - ex: if $\{A1, A2\}$ is infrequent, so is $\{A1, A2, A4\}$.
- Apriori Pruning Principle:
 - if an itemset is below the minimal support, discard all its supersets.

Example - 1

Search Space for 5 items



Example - 1 (cont.)

- Apriori enumerates and counts the support of patterns with increasing length.
- Starts looking for frequent itemsets of size 1 (F_1), assuming $minsup = 50\%$ (2 transactions)
- $C_1 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$

TID	ITEM-SET
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

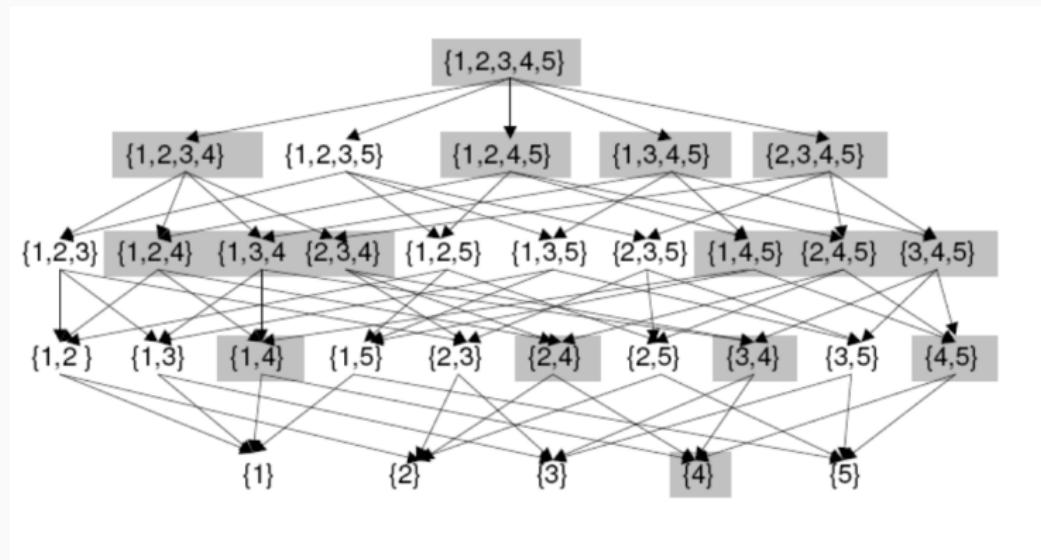


ITEM-SET	Support
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

- $F_1 = \{\{1\}, \{2\}, \{3\}, \{5\}\}$

Example - 1 (cont.)

- Filtered Search Space for 5 items (after removing item “4”)



Example - 1 (cont.)

- Looks for frequent itemsets of size 2 (F_2) from frequent itemsets of size 1 (F_1)
- Candidates $C_2 = \{\{a, b\} | \{a\} \in F_1 \wedge \{b\} \in F_1\}$
- $C_2 = \{\{1, 2\}, \{1, 3\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{3, 5\}\}$

ITEM-SET	Support
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

- $F_2 = \{\{1, 3\}, \{2, 3\}, \{2, 5\}, \{3, 5\}\}$

Example - 1 (cont.)

- Looks for frequent itemsets of size 3 (F_3) from frequent itemsets of size 2 (F_2)

- **Generation:**

$$C0_3 = \{\{a, b, c\} | \{a, b\} \in F_2 \wedge \{a, c\} \in F_2\}$$

- **Filter:**

$$C_3 = \{\{a, b, c\} | \{a, b, c\} \in C0_3 \wedge \forall x \in \{a, b, c\} S - \{x\} \in F_2\}$$

- $C_3 = \{\{2, 3, 5\}\}$

ITEM-SET	Suporte
{2,3,5}	2

- $F_3 = \{\{2, 3, 5\}\}$
- There are no frequent itemsets of size 4

Step 1 - Identifying Frequent Itemsets

- Candidate generation (**Self-Join step**)
 - generates new candidate k-itemsets based on the frequent (k-1)-itemsets found in the previous iteration.
- Candidate pruning (**Prune step**)
 - eliminates some of the candidate k-itemsets using the support-based pruning strategy.

Step 1 - Identifying Frequent Itemsets (cont.)

- Self-Join Example:

Given the size k candidates

$\{A, B, C\}$

$\{A, B, D\}$

$\{A, C, D\}$

$\{B, C, D\}$

$\{A, B, E\}$

$\{B, C, E\}$

and assuming that in each itemset the items are lexicographically sorted

- Which are the candidates of size $k + 1$?
- What is the most efficient way of finding them (without repetitions)?

Step 1 - Identifying Frequent Itemsets (cont.)

- Look for pairs of sets with the same prefix of size $k - 1$
 $\{A, B, C\}$ and $\{A, B, D\}$
- Combine both, keeping the prefix
 $\{A, B, C, D\}$
- This way
 - No frequent set is unnoticed
 - No candidate is generated more than once

Step 1 - Identifying Frequent Itemsets (cont.)

- Prune Example:

$$F_3 = \{\{A, B, C\}, \{A, B, D\}, \{A, C, D\}, \{A, C, E\}, \{B, C, D\}\}$$

$$C_4 = \{\{A, B, C, D\}, \{A, C, D, E\}\}$$

but $\{A, C, D, E\}$ can be pruned away

because $\{A, D, E\} \notin F_3$

- Note:
 - Prune maintains the completeness of the process

Step 2 - Rule Generation

- Given a frequent set $\{A, B, C, D\}$
- Which are the possible rules?
 - $\{A, B, C\} \rightarrow \{D\}$
 - $\{A, B, D\} \rightarrow \{C\}$
 - $\{A, B\} \rightarrow \{C, D\}$
- How to generate them systematically?
- How to reduce the search space?

Step 2 - Rule Generation (cont.)

- The rules are generated as follows:
 - generates all non-empty subsets s of each frequent itemset I
 - for each subset s computes the confidence of the rule $(I - s) \rightarrow s$
 - selects the rules whose confidence is higher than $minconf$

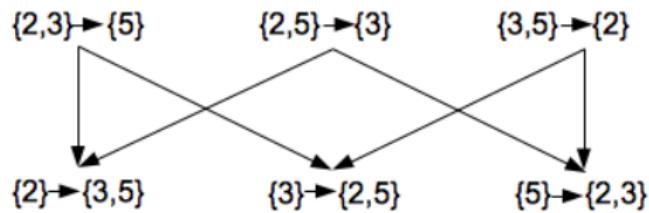
Step 2 - Rule Generation (cont.)

Consider again

Cliente (TID)	Itens (Item-set)
100	1, 3, 4
200	2, 3, 5,
300	1, 2, 3, 5,
400	2, 5,

$$\text{and } I = \{2, 3, 5\} (= F_3)$$

- Rules generated from the frequent itemset $\{2, 3, 5\}$



- Select rules $(I - a) \rightarrow a$, where $a \subseteq I$, with $minconf = 1$

$$conf((I - a) \rightarrow a) = \frac{sup(I)}{sup(I - a)}$$

Step 2 - Rule Generation (cont.)

- Rules with 1 consequent

$\{2, 3\} \rightarrow \{5\}$ (conf= 2/2)

$\{2, 5\} \rightarrow \{3\}$ (conf= 2/3) eliminated because $minconf = 1$

$\{3, 5\} \rightarrow \{2\}$ (conf= 2/2)

- Rules with 2 consequents

$\{3\} \rightarrow \{2, 5\}$ (conf= 2/3) eliminated because $minconf = 1$

- we don't need to worry about rules with item 3 in the consequent, because any rule obtained from $\{2, 5\} \rightarrow \{3\}$ will have a $conf < 2/3$

Moving items from the antecedent to the consequent never changes support and never increases confidence.

Number of DB scans

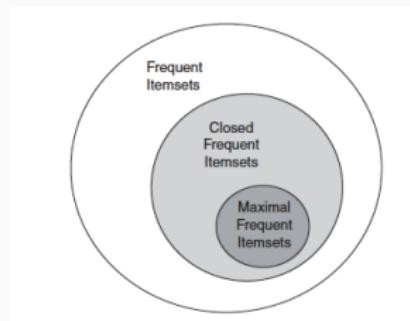
- 1 to count frequencies of C_1
- C_2 built in memory
- 2 to count frequencies of C_2
- ...
- n to count frequencies of C_n
- Rule generation does not need to scan DB
- Number of scans is n
 - if the size of the largest frequent set is n or $n - 1$

Complexity factors

- Number of items
- Number of transactions
- Minimal support
- Average size of transactions
- Number of frequent sets
- Average size of a frequent size
- Number of DB scans
 - k or $k + 1$, where k is the size of the largest frequent set

Compact Representation of Itemsets

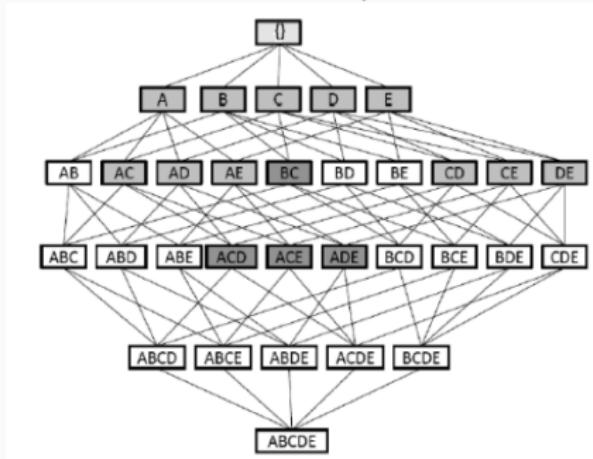
- The number of frequent itemsets produced from a transaction data set can be very large.
 - It is useful to identify a small representative set of itemsets from which all other frequent itemsets can be derived.
-
- Two such representations are:
 - maximal
 - closed



Compact Representation of Itemsets (cont.)

- s is a **closed frequent itemset** if it is a frequent itemset that has no frequent supersets with the same support.
- Example: find closed frequent itemsets with $\text{minsup} = 30\%$

TID	Itemset
1	A D E
2	B C D
3	A C E
4	A C D E
5	A E
6	A C D
7	B C
8	A C D E
9	B C E
10	A D E

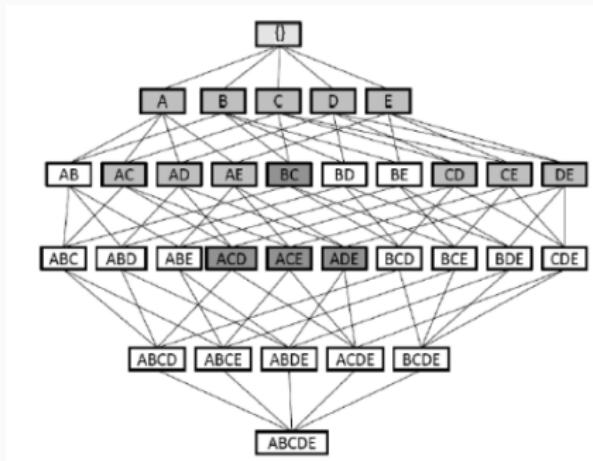


closed frequent itemsets are:
 $\{A\}$, $\{C\}$, $\{D\}$, $\{E\}$, $\{A, C\}$, $\{A, D\}$, $\{A, E\}$,
 $\{B, C\}$, $\{C, D\}$, $\{C, E\}$, $\{A, C, D\}$, $\{A, C, E\}$, $\{A, D, E\}$

Compact Representation of Itemsets (cont.)

- s is a **maximal frequent itemset** if it is a frequent itemset for which none of its supersets is frequent.
- Example: find maximal frequent itemsets with $minsup = 30\%$

TID	Itemset
1	A D E
2	B C D
3	A C E
4	A C D E
5	A E
6	A C D
7	B C
8	A C D E
9	B C E
10	A D E



maximal frequent itemsets are:

$\{B, C\}$, $\{A, C, D\}$, $\{A, C, E\}$, $\{A, D, E\}$

Compact Representation of Itemsets (cont.)

- From the maximal itemsets is possible to derive all frequent itemsets (not their support) by computing all non-empty intersections.
 - subsets of the maximal frequent itemset $\{A, C, D\}$ are frequent itemsets
 - $\{A\}, \{C\}, \{D\}, \{A, C\}, \{A, D\}, \{C, D\}$
- The set of all closed itemsets preserves the knowledge about the support values of all frequent itemsets.
 - $\{D, E\}$ is a non closed frequent itemset. What is its support?
 - As it is not closed, its support must be equal to one of its immediate supersets.
 - look for the most frequent closed itemset that contains $\{D, E\}$: $\{A, D, E\}$
 - $sup(\{D, E\}) = sup(\{A, D, E\})$
- There are algorithms that take advantage of this compact representation of frequent itemsets.

Too many rules ...

- The association rule algorithms tend to generate an excessive number of rules (for some problems, there can be thousands).
- Too many rules leads to model's interpretability lack.
- How can we reduce this number?
 - Changing the parameters: minsup , minconf
 - Restrictions on items: which items are relevant?
 - Summarization techniques: can we represent subsets of rules by a single representative rule?
 - Filter rules: improvement, measures of interest, ...

How to measure the improvement of a rule?

Improvement [Bayardo and Ag, 2000]

- **Improvement** of a rule is the minimum difference between its confidence and the confidence of any of its immediate simplifications.

$$improv(A \rightarrow C) = \min(\{conf(A \rightarrow C) - conf(As \rightarrow C) \mid As \subseteq A\})$$

- Example:

- $R_1 : \{eggs, flower, bread\} \rightarrow \{sugar\}$ ($conf = 0.505$)
- $R_2 : \{eggs, flower\} \rightarrow \{sugar\}$ ($conf = 0.5$)
- $improv(R_1)$ is at most 0.005
- with a $minimprov$ of 0.01, R_1 is excluded.

Are all the rules interesting?

- Are all the discovered patterns interesting?
- In recent years, several measures have been proposed to extract interesting patterns.
- The idea is to select a subset of rules, that somehow are more relevant.
- **Interesting rule** (Silberschatz & Tuzhilin,95)
 - Unexpected, surprising to the user
 - Measure of interest: deviation from the expected or from the initial belief
 - Useful, actionable
 - Measure of interest: estimated benefit

How to measure the interest of a rule?

- **Subjective measures:** based on user's belief in the data (ex: unexpectedness, novelty, actionability, confirm hypothesis user wishes to validate)
 - These measures are hard to incorporate in the pattern discovery task.
- **Objective measures:** based on facts, statistics and structures of patterns (ex: support and confidence), independent of the domain considered.
 - For instance, patterns that involve mutually independent items or cover very few transactions are considered uninteresting.

How to measure the interest of a rule? (cont.)

Typically

- $A \rightarrow B$ is interesting if A and B are not statistically independent
- if A and B are statistically independent, the occurrence of A does not affect the probability of occurrence of B

$$sup(A \cup B) \approx sup(A) * sup(B)$$

$$conf(A \rightarrow B) \approx conf(\emptyset \rightarrow B)$$

- $A \rightarrow B$ may have high support and confidence and still not be interesting.
 - $\{butter\} \rightarrow \{bread\}$ ($sup = 5\%$, $conf = 95\%$)
 - it is not unexpected
 - it is not useful

How to measure the interest of a rule? (cont.)

- A measure of interest should evaluate the deviation from independence.
- A rule is unexpected as it deviates from independence.
- There are different approaches to measure this deviation:
 - *lift*
 - *conviction*
 - χ^2
 - *correlation*
 - ...

Measures of Interest: limitations of support and confidence

- Assume we are interested in studying the relationship between people who drink tea and coffee.
- We summarize the preferences of 1000 people

	<i>Coffee</i>	\neg <i>Coffee</i>	
<i>Tea</i>	150	50	200
\neg <i>Tea</i>	650	150	800
	800	200	1000

- How interesting is the rule $Tea \rightarrow Coffee$?
- $sup = 150/1000 = 15\%$ and $conf = 150/200 = 75\%$
- The confidence of the rule is high, however the likelihood of a person drinking coffee regardless of drinking tea is 80%.
- Knowing that a person drinks tea actually decreases the probability of drinking coffee (from 80% to 75%).
- Thus, the rule is indeed deceitful.
- High confidence rules can be misleading.

Measures of Interest: LIFT

- **lift** is the ratio between confidence of the rule and the support of the itemset appearing in the consequent:

$$lift(A \rightarrow B) = \frac{conf(A \rightarrow B)}{sup(B)} = \frac{sup(A \cup B)}{sup(A)sup(B)}$$

- Measures the influence of A in the presence of B .
- $lift = 1$: A and B are independent ($sup(A \cup B) = sup(A)sup(B)$).
- $lift < 1$: A and B are negatively correlated.
- $lift > 1$: A and B are positively correlated.
- $lift(Tea \rightarrow Coffee) = 0.15 / (0.2 * 0.8) = 0.9375$
- negative correlation between tea and coffee drinkers.

Measures of Interest: LIFT (cont.)

- The **lift** is a measure of the deviation from a rule $A \rightarrow B$ regarding the statistical independence between the antecedent A and consequent B .
- Takes values between 0 and infinity:
 - a value close to 1 indicates that A and B often appear together
 - the occurrence of A has no effect on the occurrence of B .
 - a value smaller than 1 indicates that A and B appear less frequently than expected together
 - the occurrence of A has a negative effect on the occurrence of B , i.e. the occurrence of A is likely to lead to the absence of B .
 - a value greater than 1 indicates that A and B appear more often together than expected
 - the occurrence of A has a positive effect on the occurrence of B , i.e. the occurrence of A increases the likelihood of occurrence of B .

Measures of Interest: Conviction

- **lift** measures co-occurrence only (not implication) and is symmetric with respect to antecedent and consequent, i.e.
 $lift(A \rightarrow B) = lift(B \rightarrow A)$
- **conviction** is a measure proposed to tackle some of the weaknesses of *confidence* and **lift**.
- Unlike **lift**, **conviction** is sensitive to rule direction. It indicates the departure from independence of A and B taking into account the implication direction.
- Is inspired in the logical definition of implication and attempts to measure the degree of implication of a rule.

Measures of Interest: Conviction (cont.)

- **conviction** of a rule $A \rightarrow B$ is the ratio between
 - the expected frequency that A occurs without B , if A and B were independent
 - the observed frequency that the rule makes of incorrect predictions.
- Is the inverse **lift** of the rule $R' = A \rightarrow \neg B$.

$$\text{conviction}(A \rightarrow B) = \frac{1 - \text{sup}(B)}{1 - \text{conf}(A \rightarrow B)} = \frac{\text{sup}(A)\text{sup}(\neg B)}{\text{sup}(A \cup \neg B)}$$

Measures of Interest: Conviction (cont.)

- $\text{conviction}(A \rightarrow B) = 1$ indicates independence between A and B .
- A high value of **conviction** means that the consequent depends strongly on the antecedent.
- **conviction** increases a lot when *confidence* gets closer to 1.
- Example:
 - $\text{sup}(\text{female}) = 0.5$, $\text{sup}(\text{mother}) = 0.2$
 - $\text{conf}(\text{mother} \rightarrow \text{female}) = 1$
 - $\text{lift}(\text{mother} \rightarrow \text{female}) = 0.2 / (0.2 * 0.5) = 2$
 - $\text{conviction}(\text{mother} \rightarrow \text{female}) = (1 - 0.5) / (1 - 1) = \infty$

Improving Apriori

- Challenges of Frequent Pattern Mining
 - Multiple scans of transaction database
 - Huge number of candidates
 - Tedium workload of support counting for candidates
- Improving Apriori: general ideas
 - Reduce number of transaction database scans
 - Shrink number of candidates (*bottleneck* of Apriori)
 - Facilitate support counting of candidates
- Some methods that improve Apriori's efficiency
 - Partitioning [Savasere et al., 1995]
 - Sampling [Toivonen, 1996]
 - Dynamic Itemset Counting [Brin et al., 1997]
 - Frequent Pattern Projection and Growth (FP-Growth) [Han et al., 2004]

Association Rules: Conclusions

- GOAL: Finding associations
- Association rule mining:
 - Frequent itemsets (requires min support)
 - Association rules (requires min confidence)
 - Probabilistic implications
- One of the most used data mining tools
 - Problem: generates too much rules
 - Pattern compression and pattern selection
- Several algorithms:
 - Apriori is the most known algorithm
 - There are variants of Apriori that return exactly the same patterns!
 - Completeness: find all rules.

References

References

-  Aggarwal, C. C. (2015).
Data Mining, The Texbook.
Springer.
-  Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. I. (1996).
Fast discovery of association rules.
In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. American Association for Artificial Intelligence.
-  Agrawal, R. and Srikant, R. (1994).
Fast algorithms for mining association rules in large databases.
In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499. Morgan Kaufmann Publishers Inc.
-  Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997).
Dynamic itemset counting and implication rules for market basket data.
In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, volume 26, pages 255–264. ACM.
-  Domingo, C., Gavalda, R., and Watanabe, O. (1998).
On-line sampling methods for discovering association rules.

References (cont.)

-  Gama, J. (2016).
Association rules.
Slides.
-  Gama, J., Oliveira, M., Lorena, A. C., Faceli, K., and de Leon Carvalho, A. P. (2015).
Extração de Conhecimento de Dados - Data Mining.
Edições Sílabo, 2nd edition.
-  Han, J., Kamber, M., and Pei, J. (2011).
Data Mining: Concepts and Techniques.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.
-  Han, J., Pei, J., Yin, Y., and Mao, R. (2004).
Mining frequent patterns without candidate generation: A frequent-pattern tree approach.
Data Mining and Knowledge Discovery, 8(1):53–87.
-  Jorge, A. (2016).
Association rules.
Slides.
-  Liu, B. (2011).
Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data.
Springer, 2nd edition.

References (cont.)



Savasere, A., Omiecinski, E., and Navathe, S. B. (1995).

An efficient algorithm for mining association rules in large databases.

In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 432–444. Morgan Kaufmann Publishers Inc.



Tan, P.-N., Steinbach, M., and Kumar, V. (2005).

Introduction to Data Mining.

Addison Wesley.



Toivonen, H. (1996).

Sampling large databases for association rules.

In *Proceedings of the 22th International Conference on Very Large Data Bases*, VLDB '96, pages 134–145. Morgan Kaufmann Publishers Inc.



Torgo, L. (2017).

Data Mining with R: Learning with Case Studies.

Chapman and Hall/CRC, 2nd edition.

introduction to classification

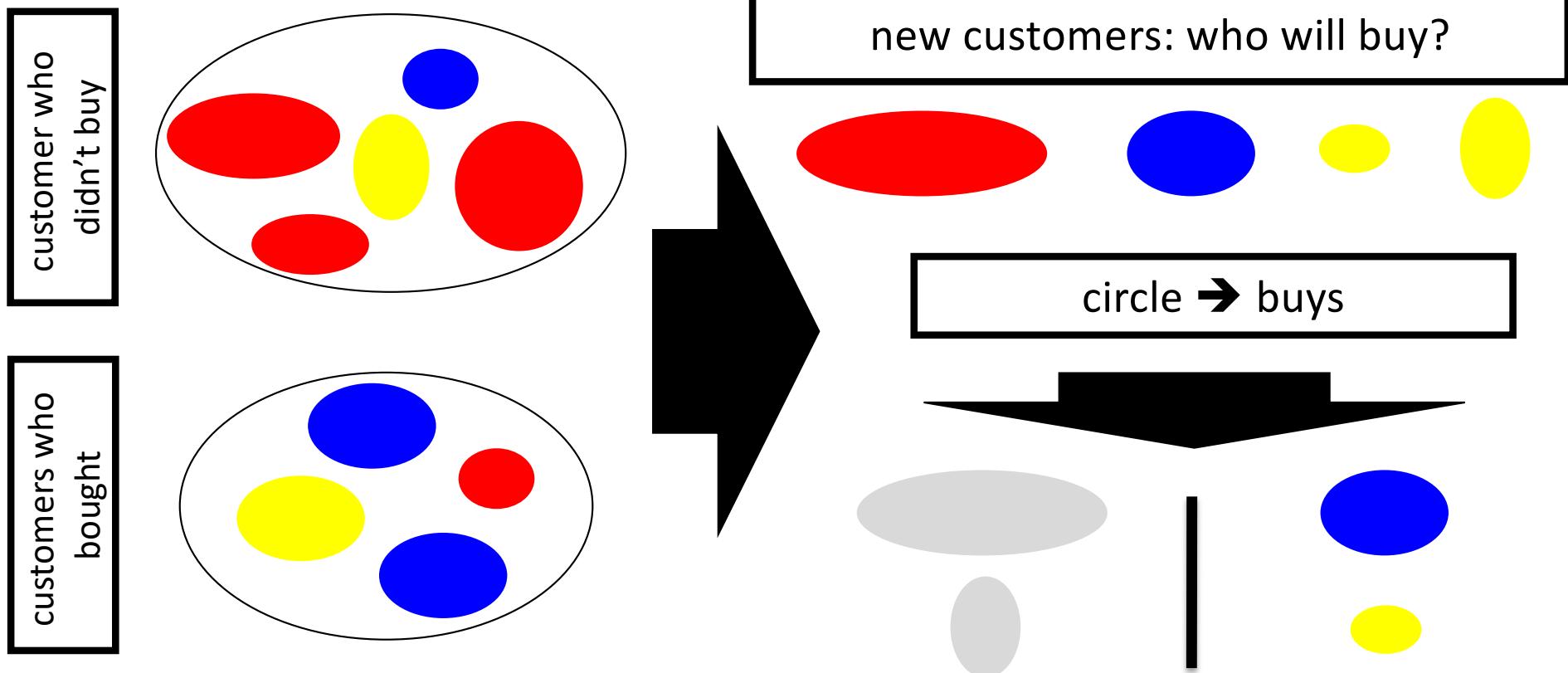
carlos soares

(csoares@fe.up.pt)

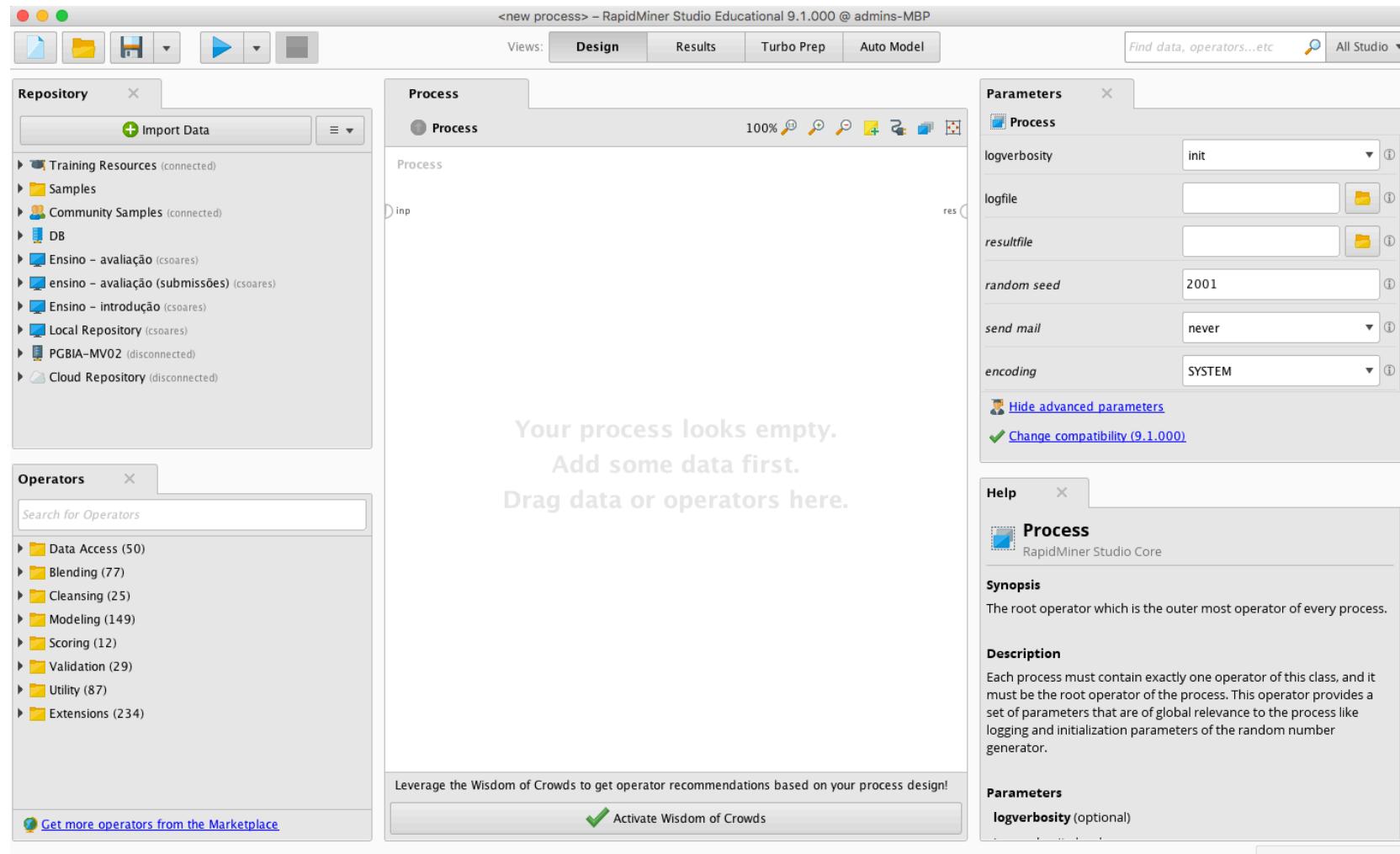
[including materials kindly provided by Alípio Jorge and adapted
from David Sontag, Luke Zettlemoyer, Carlos Guestrin and Andrew
Moore]



predictive: classification for targeting

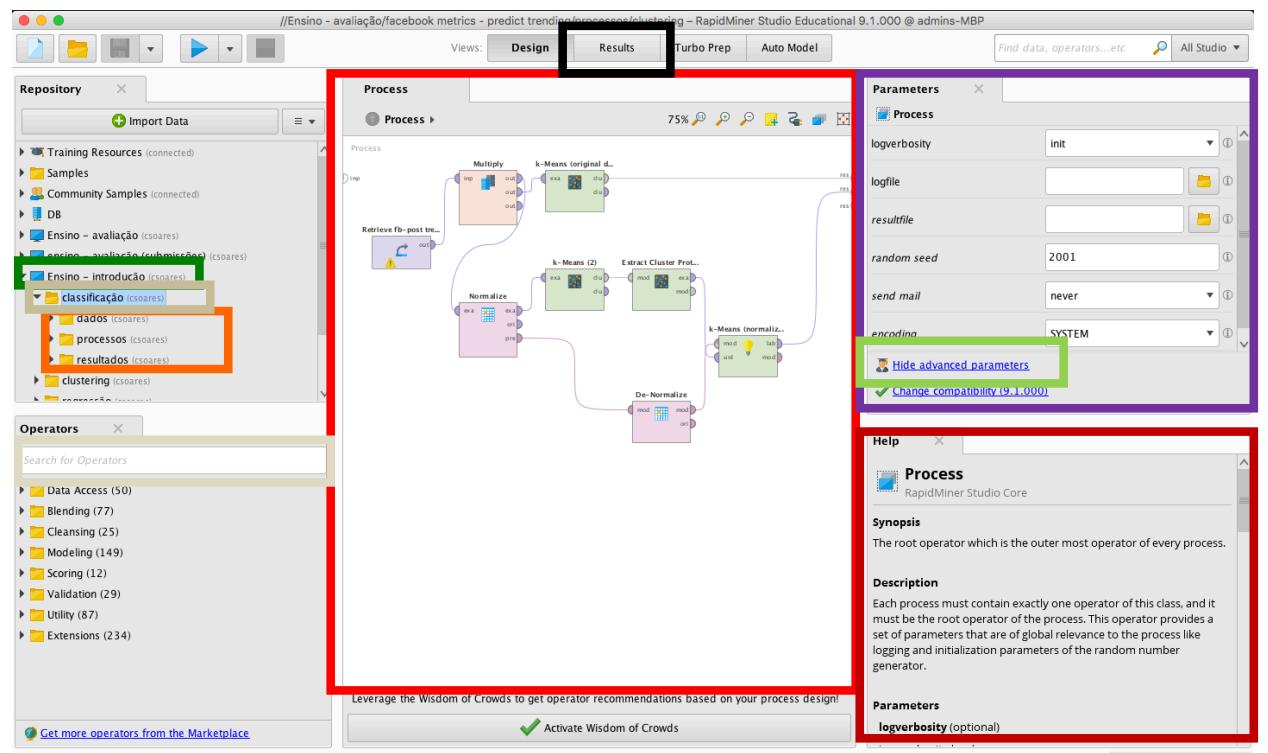


but, first, introduction to rapidminer



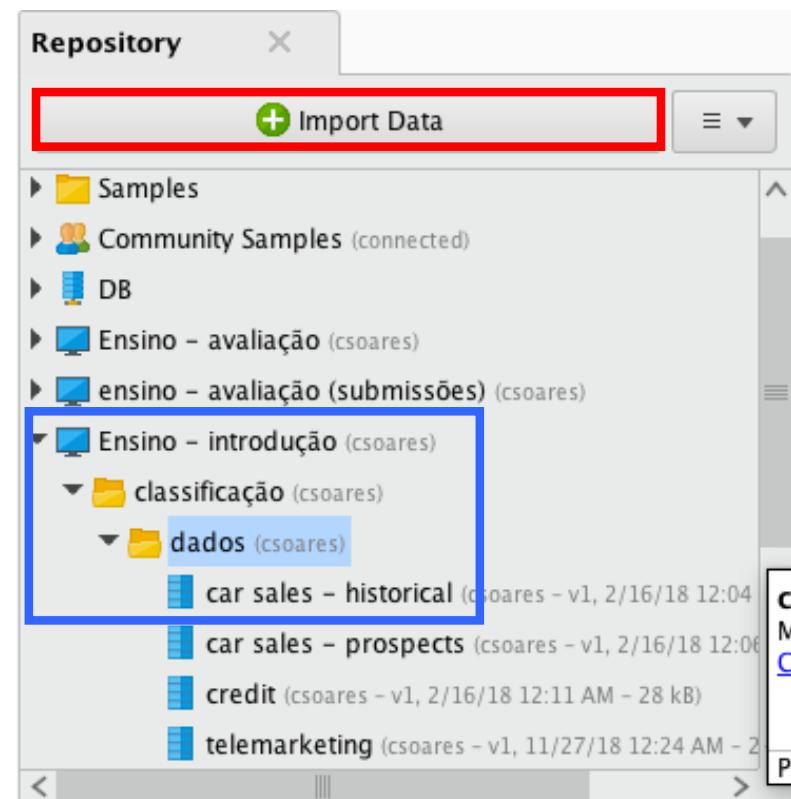
rapidminer projects

- workflow of operators
 - with parameters
 - including advanced ones
 - and help easily available
 - searchable
- repository of folders
 - e.g. repository = company
 - ... folder = project
- folders also used to organize projects
- results presented in a different view



... need data

- data are **imported** to rapidminer
 - many different formats accepted
 - ... including databases
- ... wizards available
 - use carefully!
- ... and **stored in a folder**



CLASSIFICATION

classification for campaign optimization

[fonte: ferrari](#)



- campaign to promote new vehicle
 - (large) list of prospects
 - invitations for test-drive
 - gifts
 - free phone line (800) for enquiries/reservations
- goal
 - reduce costs
 - maximize returns
- strategy
 - analyse response to previous campaigns
 - stored in a database
 - build customer relating customer characteristics and response
 - apply model to prospects
 - invite prospects selected by the model
 - [who bought last car more than 4 years ago]

data for classification

- prospects
 - customers who didn't buy a car in the last 4 years
- results from previous campaigns
 - customers who were contacted and their response

would like to predict

target (or dependent) variable

independent variable (or attribute)

Comprou	Idade	Rendimento	Ag.fam	Vendas anteriores	Última Venda
	41	50000	2	1	0
	39	68000	2	0	30000
	58	61000	4	0	0
	26	25000	3	0	0
	21	50000	1	1	20000
	38	43000	2	0	0
	44	43000	4	1	47000
	27	47000	2	1	21000
	70	23000	2	0	25000

Comprou	Idade	Rendimento	Ag.fam	Vendas anteriores	Última Venda
não	37	49000	2	1	42000
sim	43	68000	3	0	0
sim	42	61000	4	0	0
sim	26	52000	2	0	0
sim	40	64000	1	1	21000
sim	38	52000	1	0	0
sim	45	43000	4	1	47000
sim	35	45000	2	1	34000
não	39	43000	2	0	0
sim	31	55000	3	1	46000
sim	34	57000	3	1	52000
não	38	44000	4	0	0
não	34	68000	2	1	33000
...

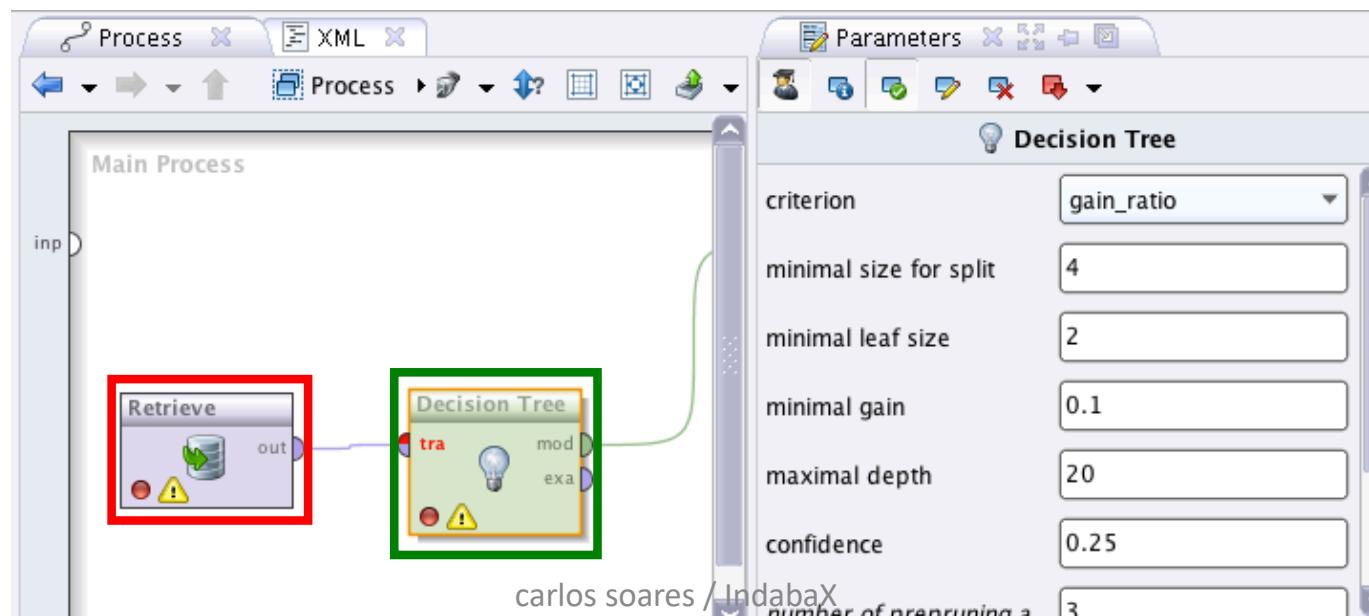
already known

exercise: marketing campaign

- predict which customers will accept invitation for test drive
 - business goal
 - efficient and effective use of sales resources
 - data available at the data.xlsx
 - worksheets
 - “car sales (historical data)”
 - “car sales (prospects)”
 - variables
 - target
 - bought?
 - features
 - age, income, family size, cars bought previouslt and value of last purchase
 - tool
 - RapidMiner

exercise: classification model in rapid miner

- load data into repository
 - target variable is a *label*!!
- load **data** from repository into workspace
- apply **decision tree** algorithm
 - e.g. operators → modeling → classification and regression → tree induction → decision tree
- analyse model



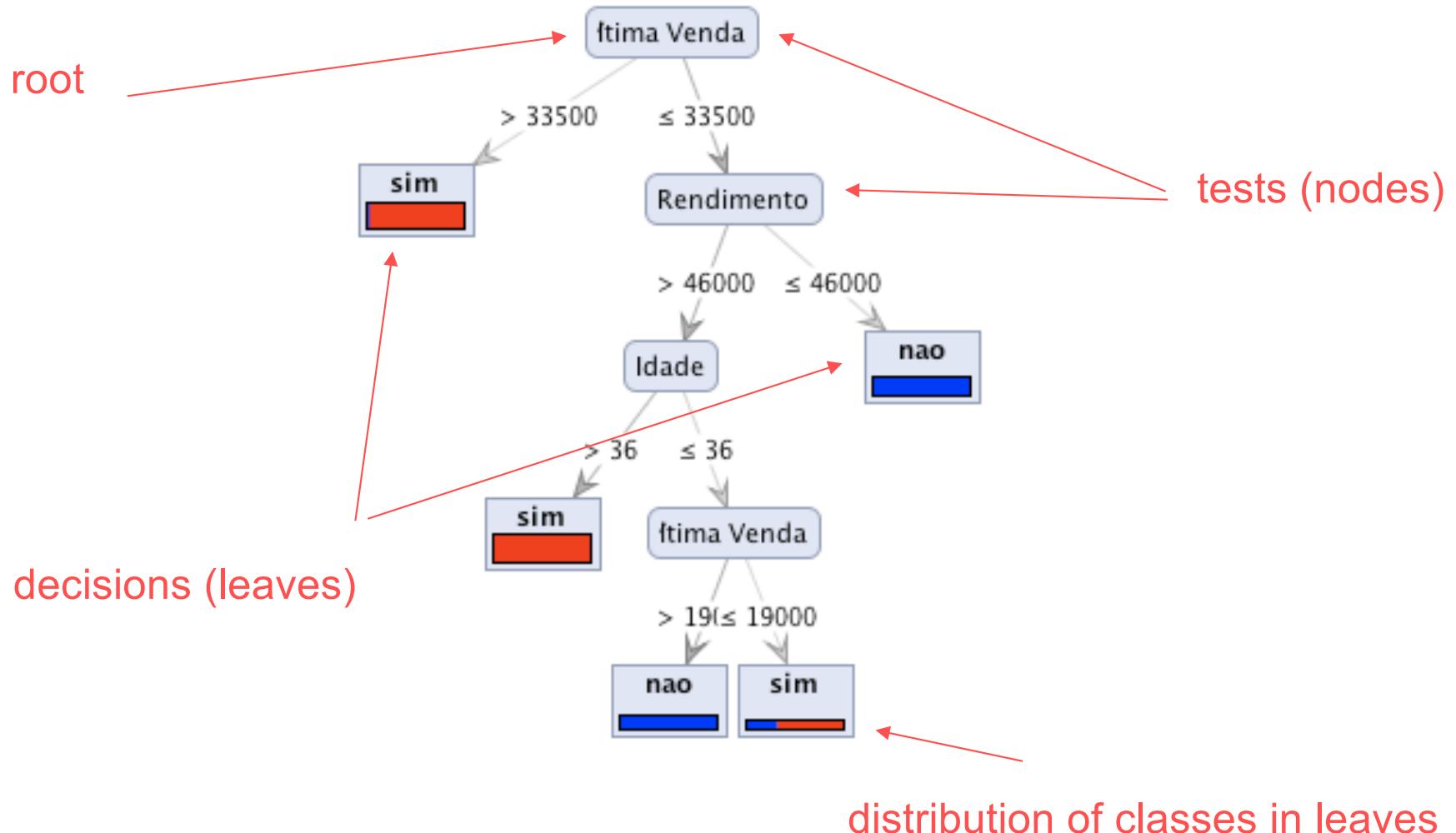
gps



- learn decision tree
- use decision tree
 - interpretation
 - application to new examples

fonte: <http://www.flickr.com/photos/emina2492/2638248645/>

model: classification tree (or decision tree)

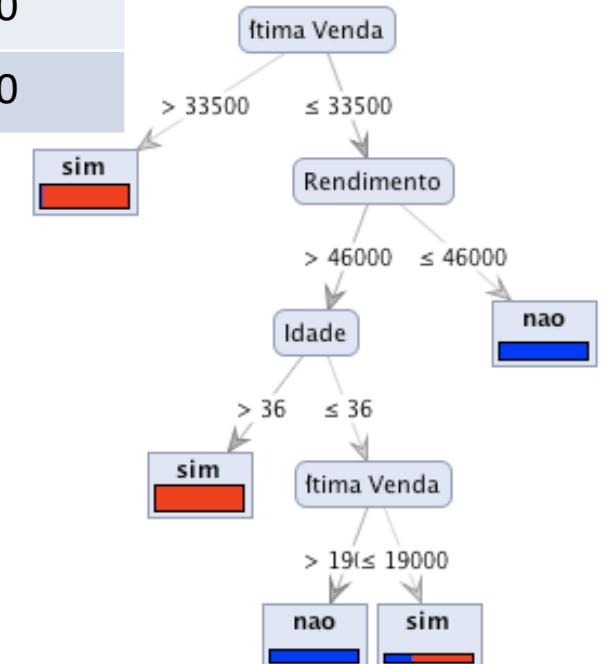


classify new examples

- prospects list

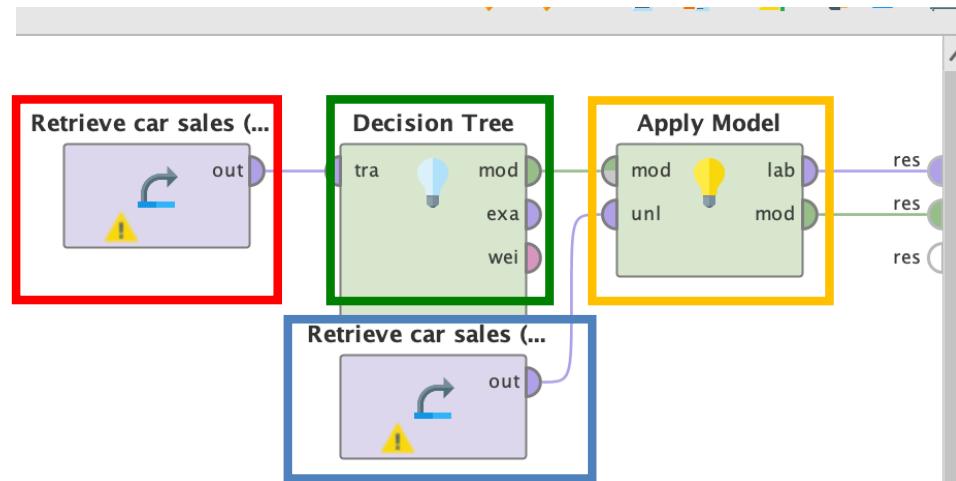
age	income	family size	previous sales	last sale
28	39.000	2	0	0
39	52.000	4	1	17.000
29	42.000	4	1	40.000

- class of the leaf each example is assigned by the tree?
 - i.e. predict...



classify new examples with rapidminer

- load new data into repository
 - target variable is a label!!
 - ... even for the new data
- load **labelled data** from repository into workspace
 - just drag it!
- apply **decision tree algorithm**
 - operator: decision tree
- load **unlabelled data** from repository into workspace
- **apply** decision tree model to the new data
 - operator: Apply Model



predictions

- **predictions** made by the model
- ... and **probability** of each class

Row No.	Bought?	prediction(...)	confidence(...)	confidence(...)	Age	Income	Family size	Cars boug...	Value of la...
1	?	nao	0.642	0.358	41	50000	2	1	0
2	?	sim	0.283	0.717	39	68000	2	0	30000
3	?	nao	0.524	0.476	58	61000	4	0	0
4	?	nao	0.935	0.065	26	25000	3	0	0
5	?	nao	0.869	0.131	21	50000	1	1	20000
6	?	nao	0.758	0.242	38	43000	2	0	0
7	?	sim	0.067	0.933	44	43000	4	1	47000
8	?	nao	0.704	0.296	27	47000	2	1	21000
9	?	nao	0.847	0.153	70	23000	2	0	25000

classification: applying a model to new cases

responses to previous campaigns

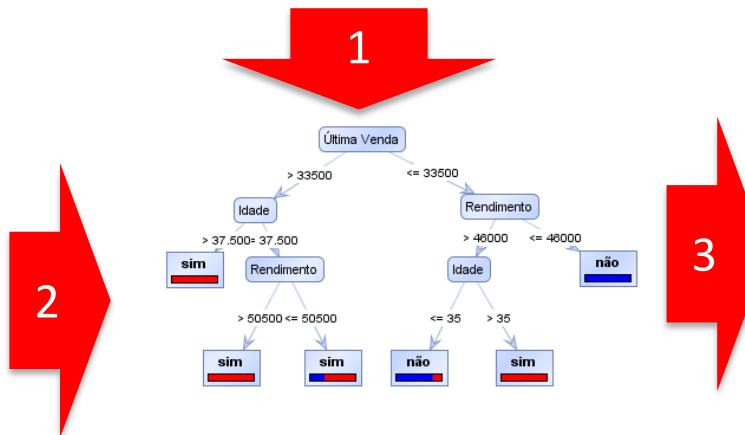
known responses (class)

Comprou	Idade	Rendimento	Ag.fam	Vendas anteriores	Última Venda
não	37	49000	2	1	42000
sim	43	68000	3	0	0
sim	42	61000	4	0	0
sim	26	52000	2	0	0
sim	40	64000	1	1	21000
sim	38	52000	1	0	0
sim	45	43000	4	1	47000
sim	35	45000	2	1	34000
não	39	43000	2	0	0

prospects

	A	B	C	D	Vendas
1	Comprou	Idade	Rendimento	Ag.fam	Vendas
2		41	50000	2	
3		39	68000	2	
4		58	61000	4	
5		26	25000	3	
6		21	50000	1	
7		38	43000	2	
8		44	43000	4	
9		27	47000	2	
10		70	23000	2	

unknown responses (class)



row no.	Comprou	prediction(...)	confidence(...)	confidence(...)	Idade
1	?	sim	0	1	41
2	?	sim	0	1	39
3	?	sim	0	1	58
4	?	não	1	0	26
5	?	não	0.818	0.182	21
6	?	não	1	0	38
7	?	sim	0	1	44
8	?	não	0.818	0.182	27
9	?	não	1	0	70

predictions by the model

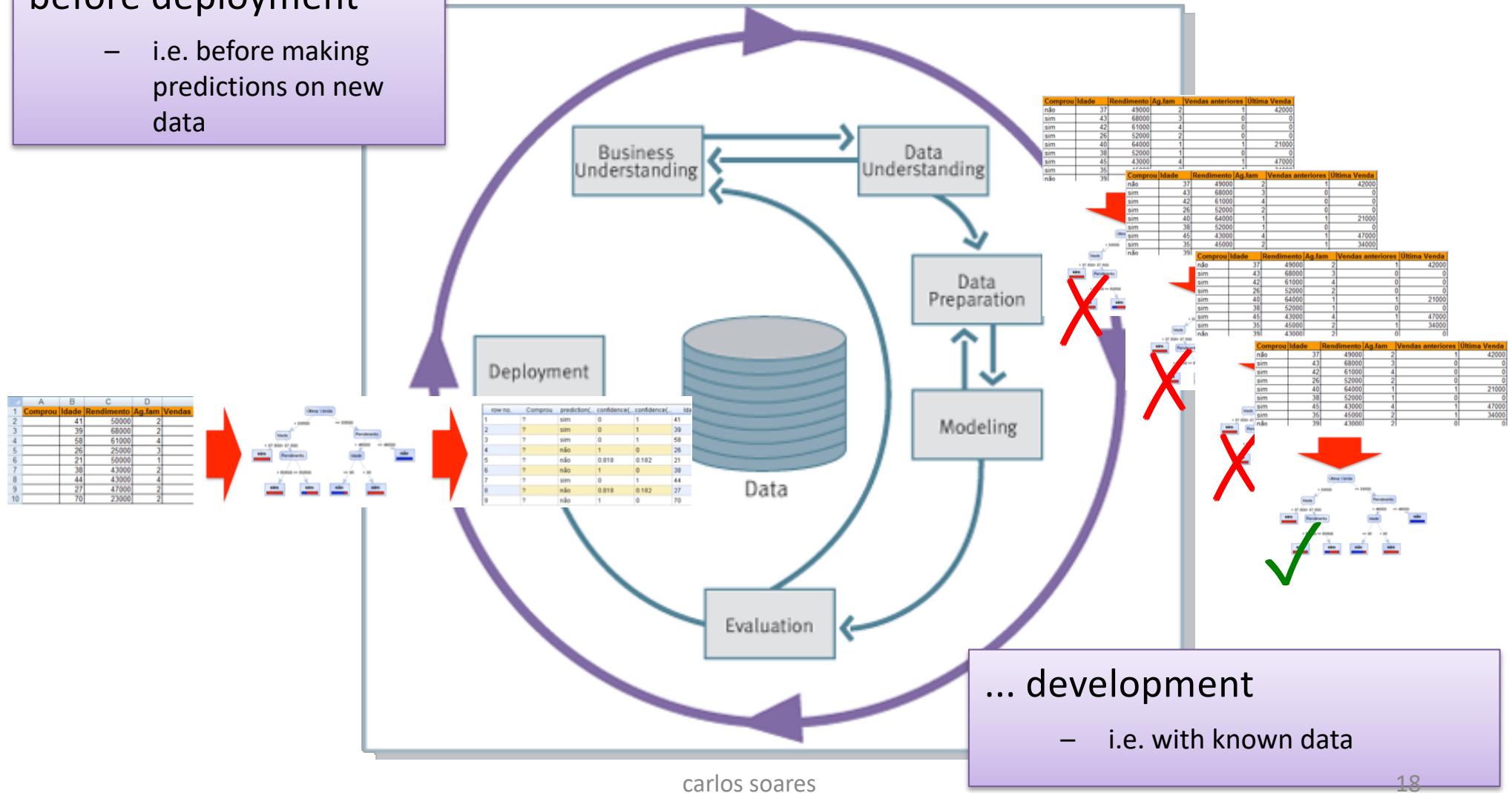
would you use the decisions proposed by this model?

CLASSIFIER EVALUATION

model development

before deployment

- i.e. before making predictions on new data



how good are the predictions?

- confusion matrix
 - prediction vs reality
 - number of right answers on the main diagonal
 - sum of the array is the total number of examples
- error rate
 - percentage/proportion of cases where the model misses
 - e.g. $(2 + 1)/(5 + 1 + 2 + 29) = 8.1\%$

	truth: no	truth: yes
prediction: no	5	1
prediction: yes	2	29

evaluation measures

- multiple measures can be computed from the confusion matrix, including...

	truth: no	truth: yes
prediction: no	TN	FN
prediction: yes	FP	TP

$\frac{FP}{FP + TN}$	False positive rate (FPR) = 1-TNR	$\frac{TP}{TP + FP}$	Positive predictive value (PPV), also known as precision
$\frac{FN}{TP + FN}$	False negative rate (FNR) = 1-TPR	$\frac{TN}{TN + FN}$	Negative predictive value (NPV)
$\frac{TP}{TP + FN}$	True positive rate (TPR), also known as recall or sensitivity	$\frac{TP + TN}{TP + TN + FP + FN}$	Accuracy
$\frac{TN}{TN + FP}$	True negative rate (TNR), also known as specificity	$\frac{2}{1/precision + 1/recall}$	F1-measure

is the model any good at all?

	truth: no	truth: yes
prediction: no	5	1
prediction: yes	2	29

- model error: $3/37 = 8.1\%$
- **baseline**
 - simplest model that can be obtained from the data

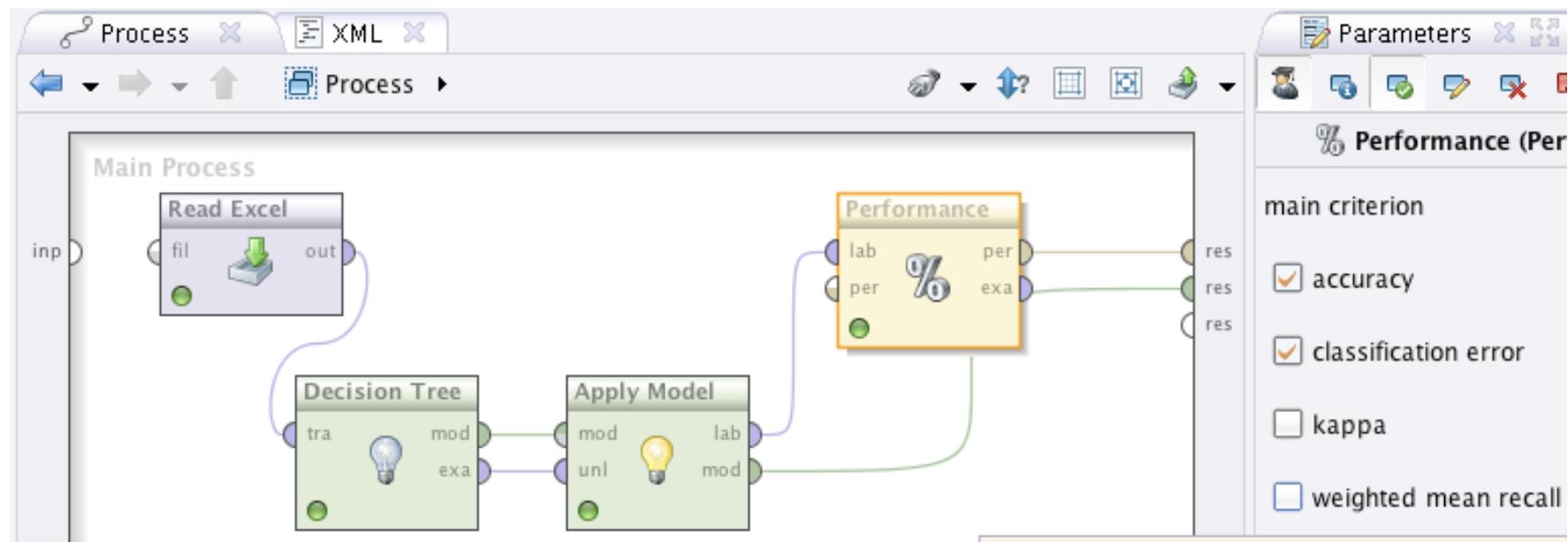
	truth: no	truth: yes
class distribution	7	30

most “popular” choice

- ... with error: $7/37 = 18,9\%$
- so, should we use the model?

exercise I: marketing campaign

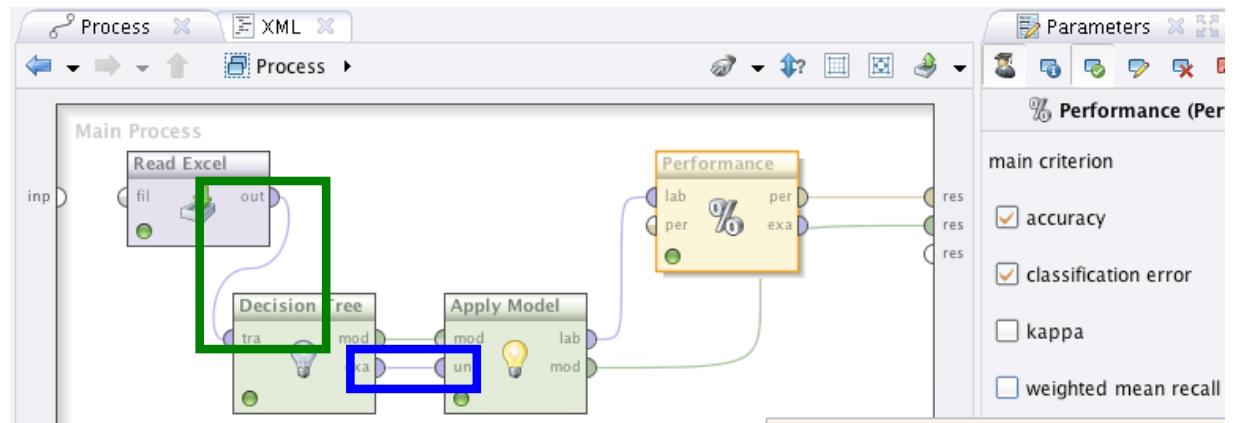
- evaluate decision tree
 - operator: performance (classification)



- doesn't this feel strange?

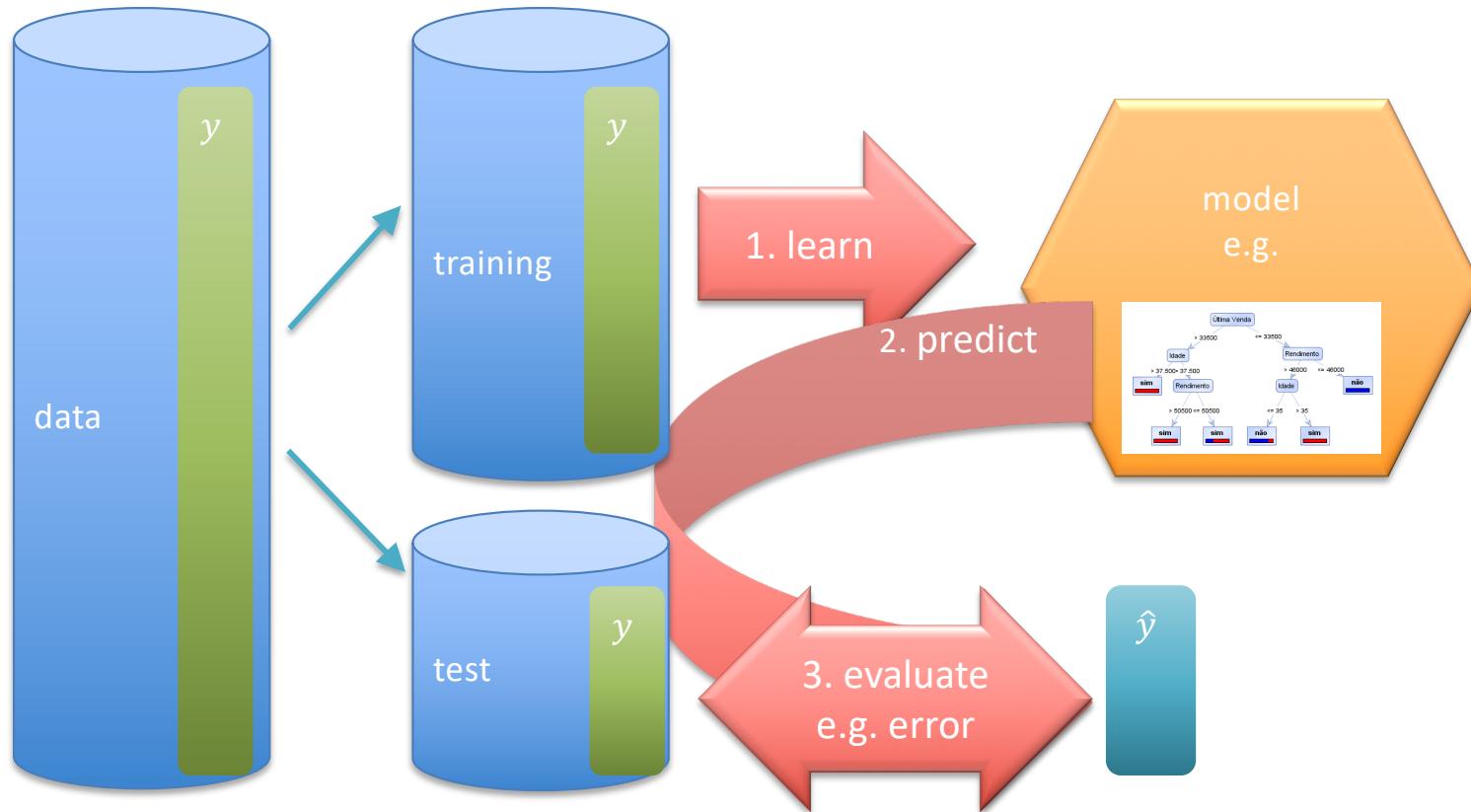
what is the value of the model?

- goal: apply the model to **new** cases
- but, so far, same data to
 - **train** model
 - ... and **evaluate it**



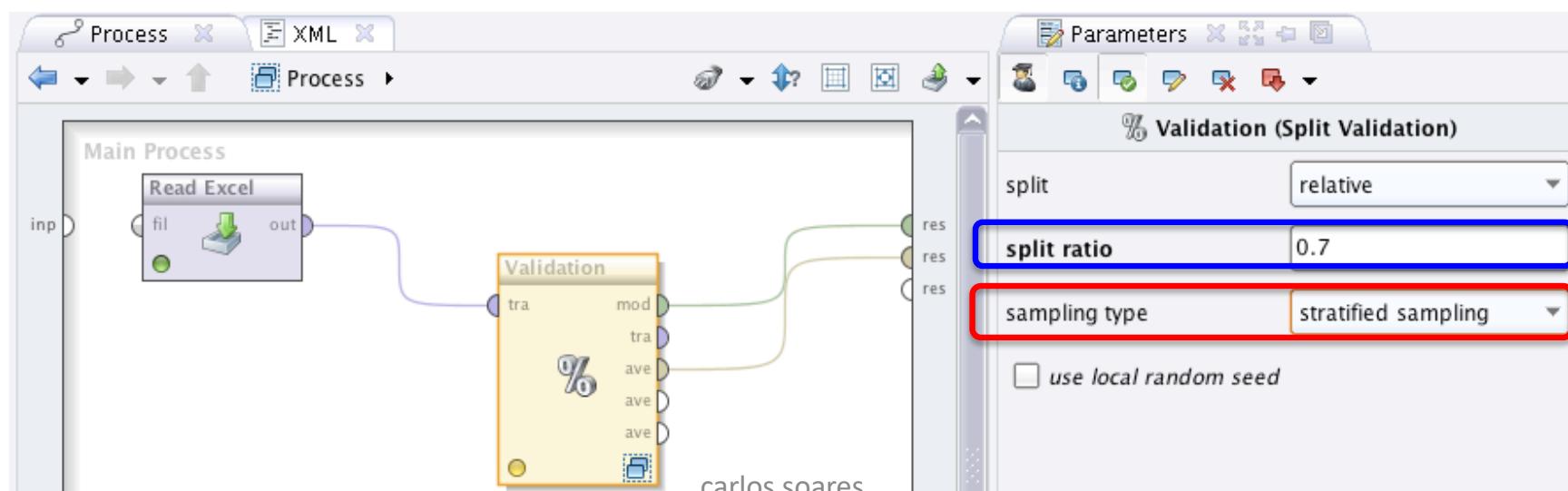
- evaluation with training data
 - it's easy(ier) to make predictions in cases you already observed
 - assumes that future cases will be equal to those of training
 - similar to giving an exam with the same problems that were solved in the last class
 - unreliable estimator of model behavior in new examples

evaluation methodology: do not forget!



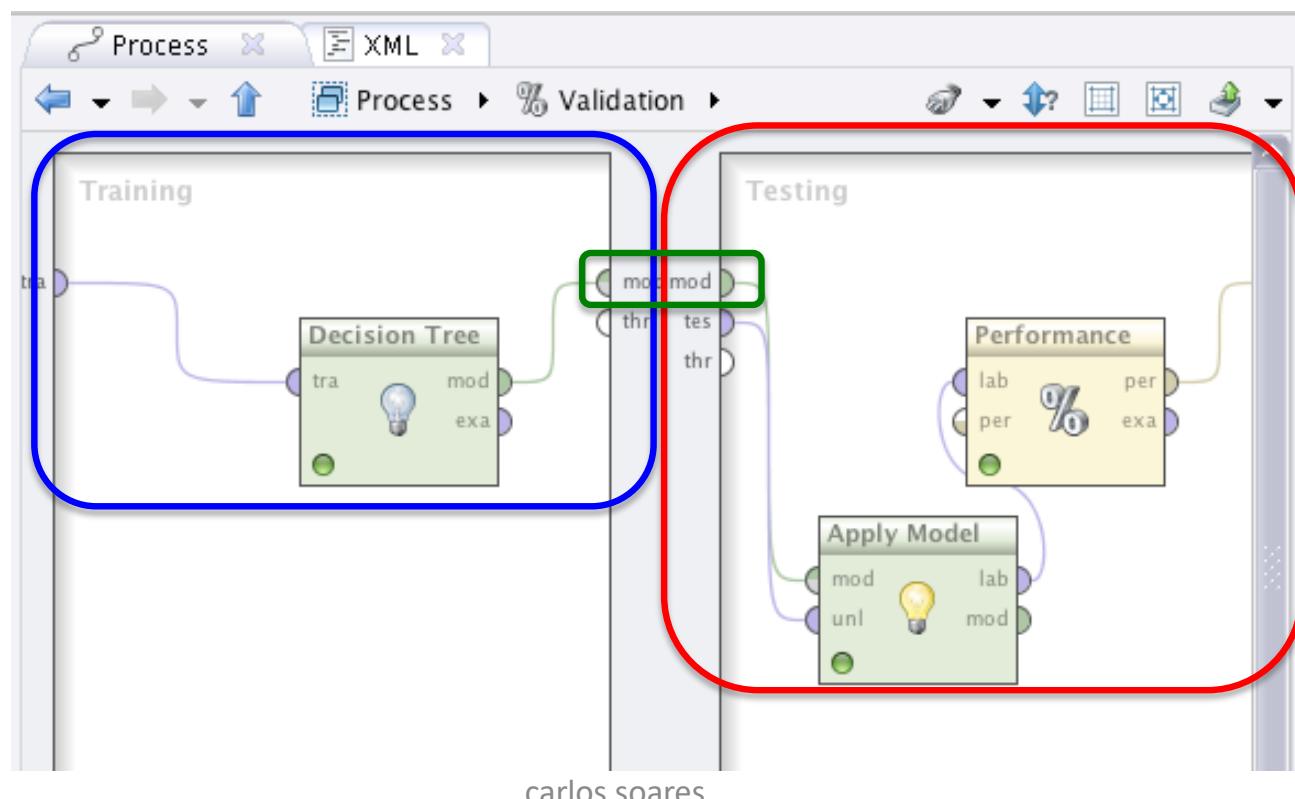
exercise II: marketing campaign (1/2)

- operator split validation
- sub-process
 - operator that groups operators
- distribute data randomly between the training and test sets
 - ensuring the same class proportion
- proportion
 - 70% of the cases for training
 - 30% of the cases for testing



exercise II: marketing campaign (2/2)

- split validation
 - different operations for **training** data and for **test** data
 - **model** obtained on the train side is passed on to the test side



DECISION TREES

- how the algorithm for induction of decision trees works
- overfitting

learning a decision tree (1 and 2/5)

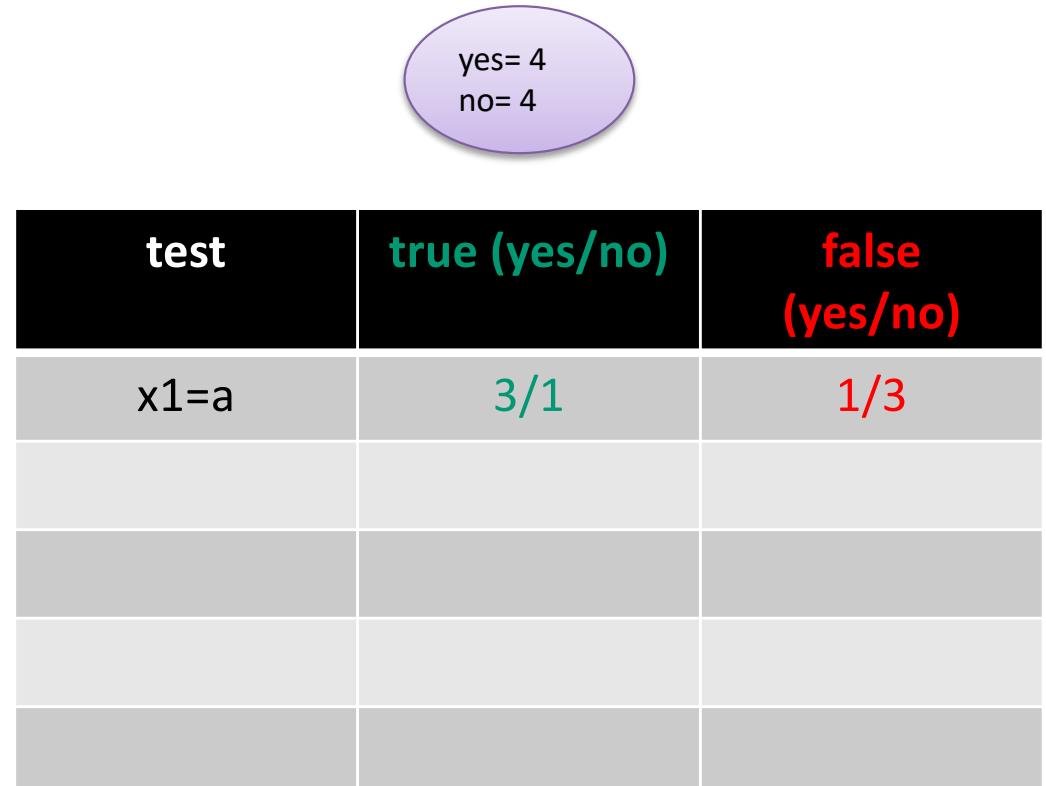
x1	x2	class
a	p	yes
a	q	no
a	p	yes
b	q	no
c	p	no
a	p	yes
c	p	yes
b	p	no

yes= 4
no= 4

1. we have a set of labelled examples
 - the target variable indicates the class of each case (e.g. yes, no)
 - on the root knot we have all the cases
2. if all the examples are of the same class, we stop

learning a decision tree (3/5)

x1	x2	class
a	p	yes
a	q	no
a	p	yes
b	q	no
c	p	no
a	p	yes
c	p	yes
b	p	no

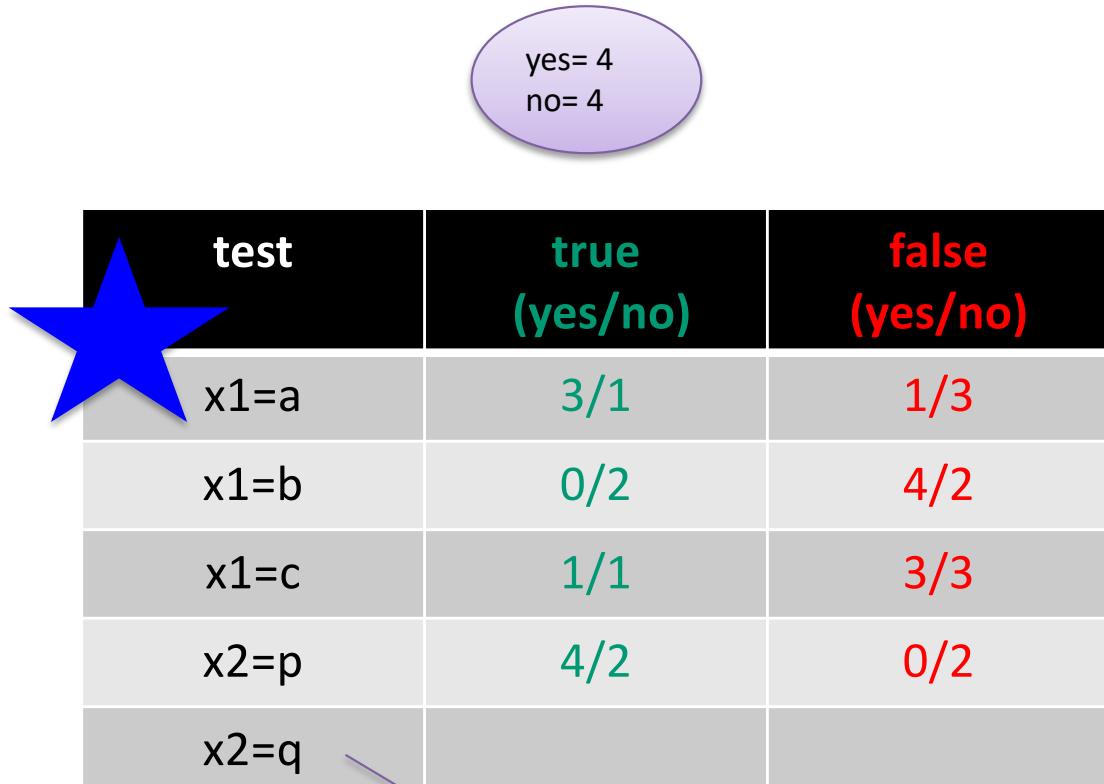


3. otherwise, we will divide (split) the examples in the root

- so that the classes are well separated
- each test is of variable type = value, or variable > value

learning a decision tree (3/5 – cont'd)

x1	x2	class
a	p	yes
a	q	no
a	p	yes
b	q	no
c	p	no
a	p	yes
c	p	yes
b	p	no



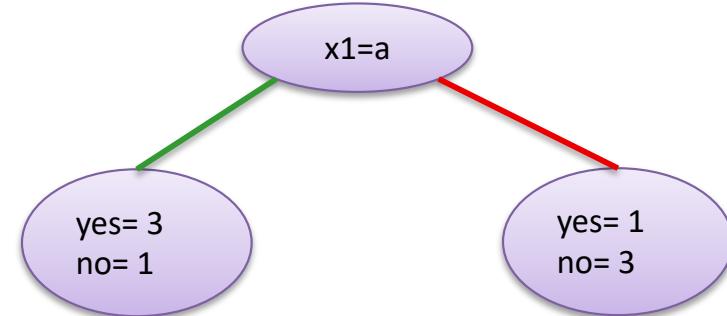
3. otherwise, we will divide (split) the examples in the root

- so that the classes are well separated
- each test is of variable type = value, or variable > value

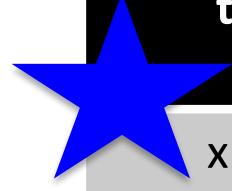
unnecessary

learning a decision tree (4/5)

x1	x2	class
a	p	yes
a	q	no
a	p	yes
b	q	no
c	p	no
a	p	yes
c	p	yes
b	p	no



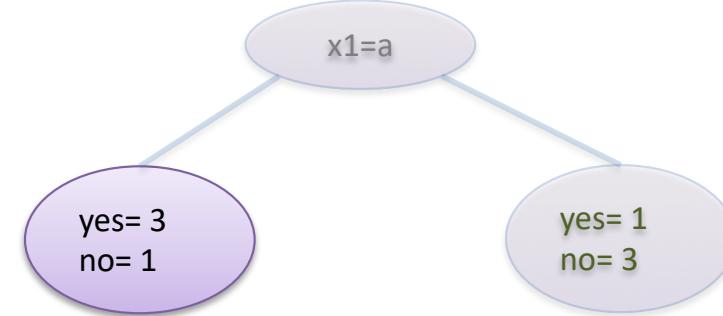
4. create two descendant nodes according to the selected test



test	true (yes/no)	false (yes/no)
$x_1 = a$	3/1	1/3

learning a decision tree (5/5)

x1	x2	class
a	p	yes
a	q	no
a	p	yes
b	q	no
c	p	no
a	p	yes
c	p	yes
b	p	no



test	true (yes/no)	false (yes/no)
$x_1 = a$	3/1	1/3
$x_2 = p$	3/0	0/1

- We repeat the process for the set of examples in each of these descendant nodes

unnecessary

splits: the **good**, the **bad** and the **ugly**

x1	x2	class
a	p	yes
a	q	no
a	p	yes
b	q	no
c	p	no
a	p	yes
c	p	yes
b	p	no

too good to be true?

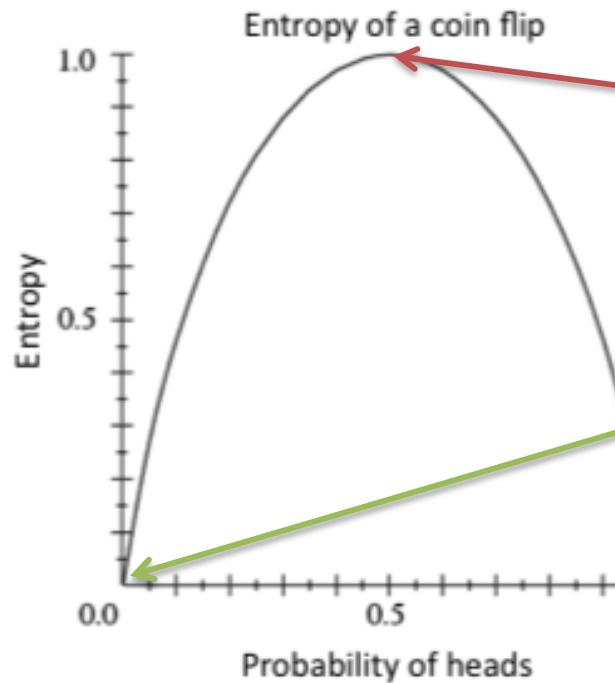
good or bad?...

test	true (yes/no)	false (yes/no)
x1=a	3/1	1/3
x1=b	0/2	4/2
x1=c	1/1	3/3
x2=p	4/2	0/2
x2=q		

is this always bad?

entropy as diversity

- Entropy $H(Y)$ of a random variable Y



test	true (yes/no)	false (yes/no)
x1=a	3/1	1/3
x1=b	0/2	4/2
x1=c	1/1	3/3
x2=p	4/2	0/2
x2=q		

$$H(Y) = - \sum_{i=1}^k P(Y = y_i) \log_2 P(Y = y_i)$$

... the bigger the decrease in diversity, the better

x1	x2	class
a	p	yes
a	q	no
a	p	yes
b	q	no
c	p	no
a	p	yes
c	p	yes
b	p	no

test	true (yes/no)	false (yes/no)
x1=a	3/1	1/3

i.e. d(current set)

>>

d(left branch) + d(right branch)

e.g. information gain

$$IG(X) = H(Y) - H(Y | X)$$

numerical attributes

...	x3	class
1	yes	
3	no	
2	yes	
3	no	
9	no	
5	yes	
5	yes	
3	no	



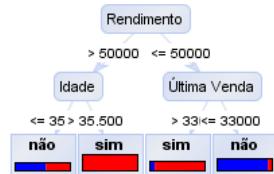
...	x3	class
1	yes	
2	yes	
3	no	
3	no	
3	no	
5	yes	
5	yes	
9	no	

test	true (yes/no)	false (yes/no)
$x3 < 1.5$	1/0	3/4
$x3 < 2.5$	2/0	2/4
$x3 < 4$	2/3	2/1
$x3 < 7$	4/3	0/1

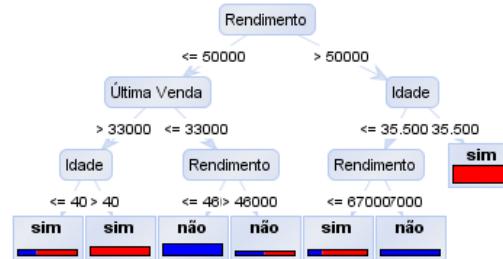
- only splits between examples of different classes should be considered
 - $x3 < 1.5$ cannot be better than $x3 < 2.5$

overfitting

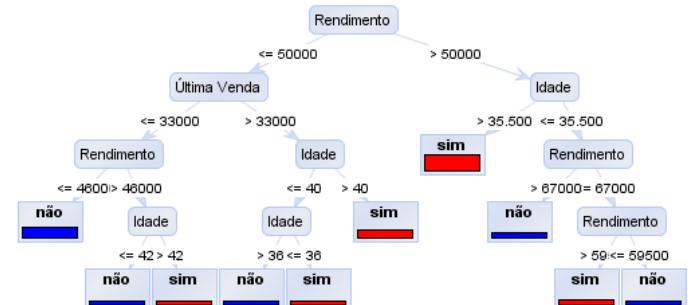
- trees obtained with different values of “minimum leaf size”
 - 4, 2 and 1



error (train)=18,18



error (train)=9,09%

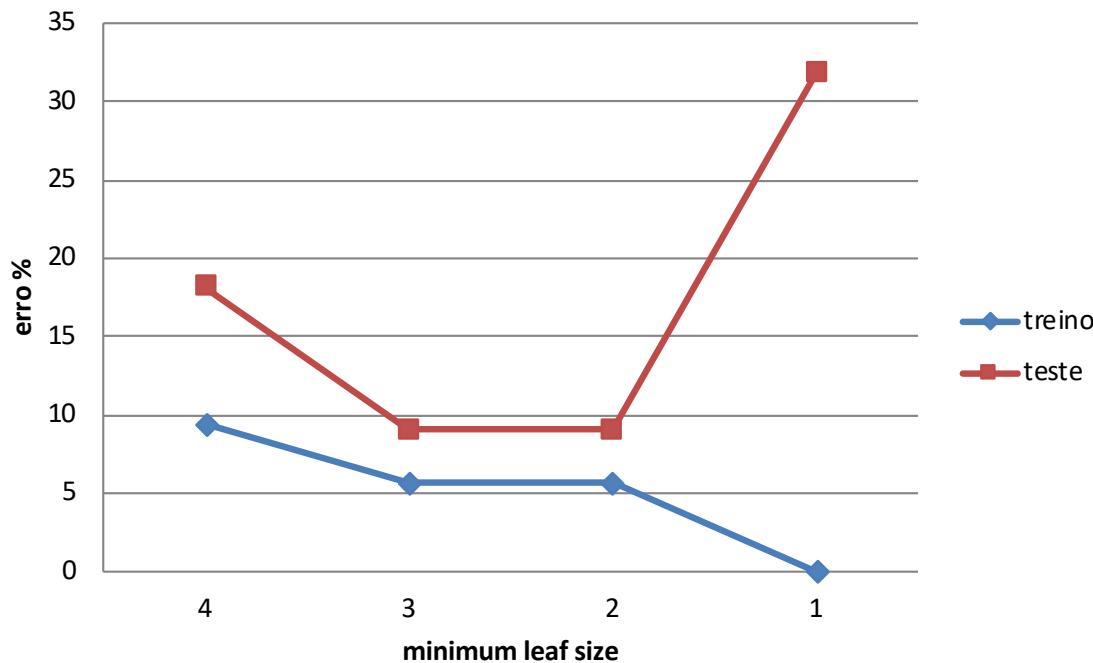


error (train)=0,00%

- However
 - decreasing the training error does not mean that the model is better on new cases
 - a model too adjusted to training cases is overfitted
 - an overfitted model generalizes poorly

overfitting: experiment

- how does the error of the DT vary in test data?
 - vary the minimum leaf size between 4 and 1



- training error always goes down
- test error begins by descending but then rises again
 - an overadjusted AD generalizes poorly

CLASSIFICATION FOR SCORING

plan

- binary classification
- evaluation in binary classification

classification: b&w or gray?

- direct application of the model splits examples into classes
 - eg. good and bad customers/buys or doesn't buy
- not suitable for all problems
 - list of 1000 prospects but send only to the 200 with the highest probability of buying
 - ... what if model selects only 30
 - ... or 300?
- *scoring*: use estimated probability of buying to order cases
 - select 200 with the highest probability ... or score
- score also provides information about (um)certainty of prediction

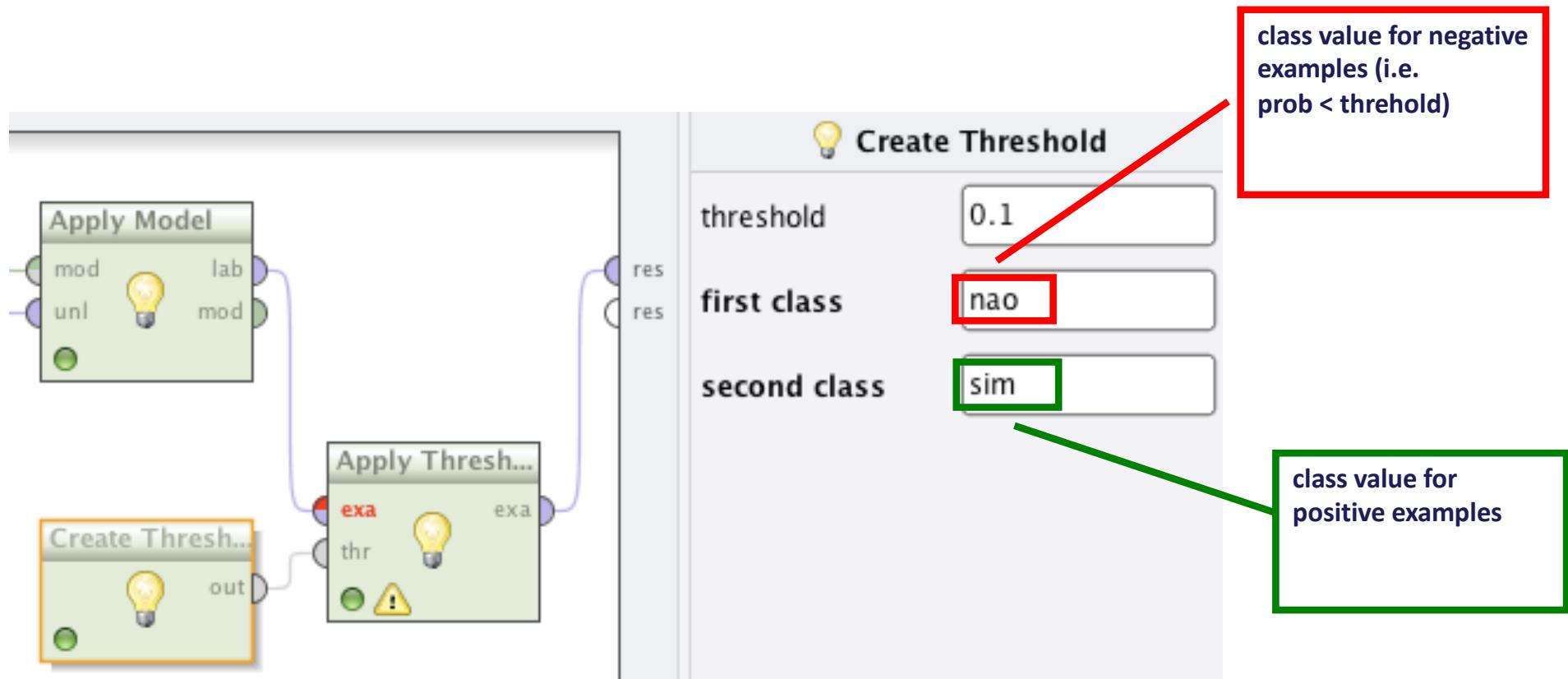


fonte: <http://www.flickr.com/photos/backpackphotography/3354435787/>

where to cut?

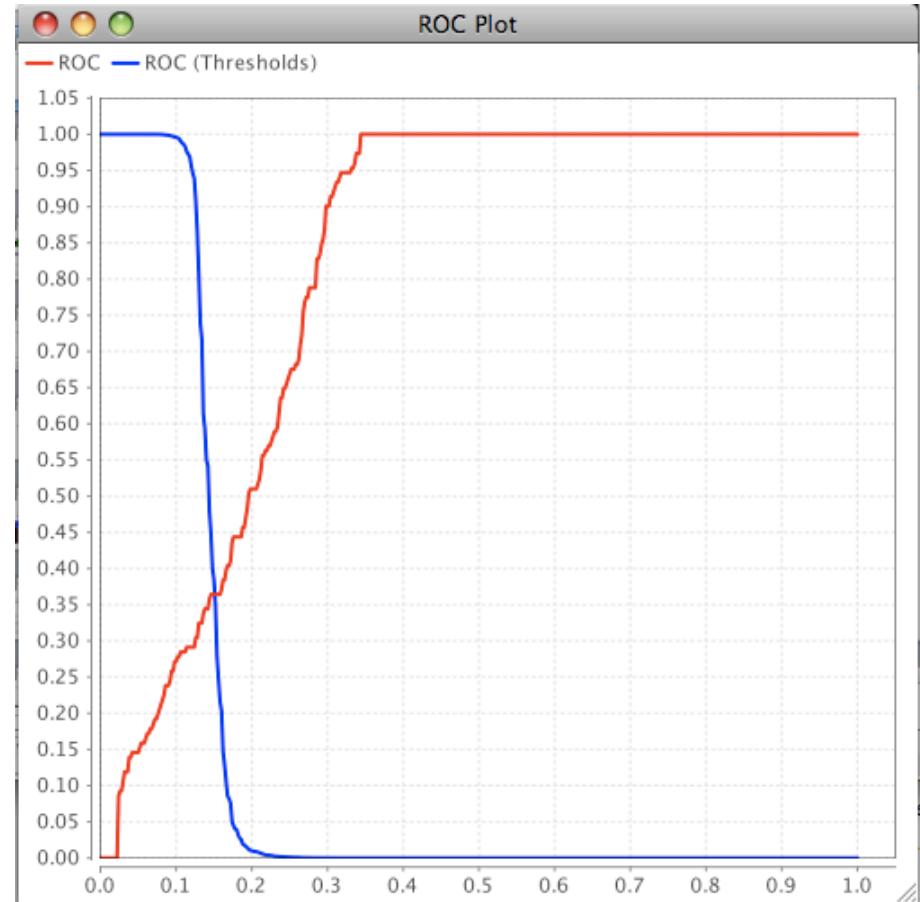
- imposed by available resources
 - n cases
- is n a suitable number?
 - maybe too many “bad” customers...
- arbitrary threshold
 - by default, class = yes if $\text{Prob}(\text{yes}) > 0.5$
- ... but which is the right value?
 - eg. important to find all “yes”
 - ... send if $\text{Prob}(\text{sim}) > 0.3$

... in RM



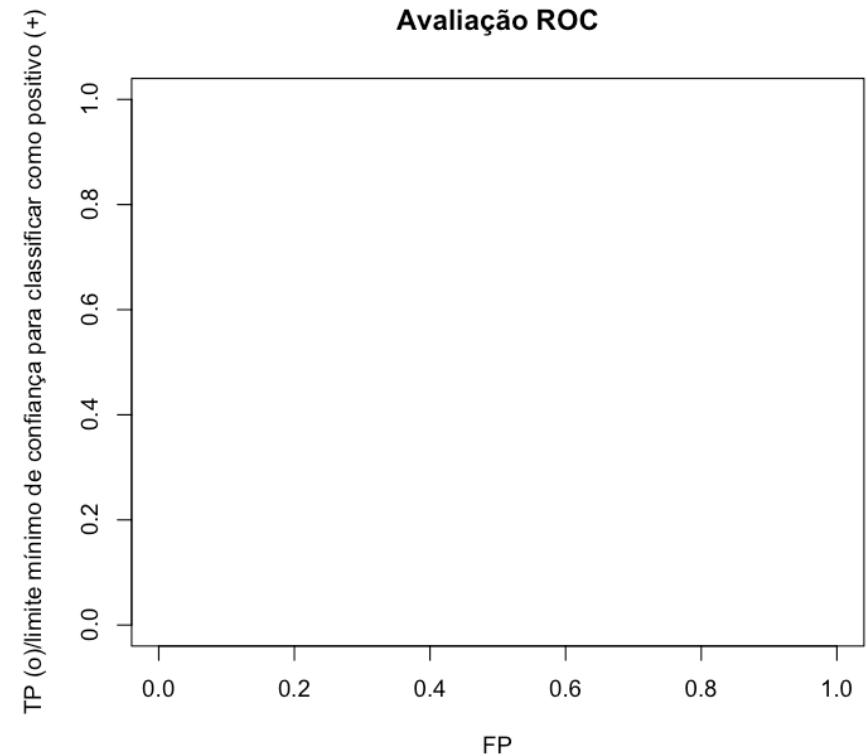
evaluate scoring models

- sort prediction by increasing order of belonging to positive class
 - $P(\text{sim} \mid \text{features})$
- ROC analysis
 - *Receiver Operating Characteristic*
 - visualize proportion TP vs. FP
 - ... threshold
 - only rapidminer
 - ... to find best compromise

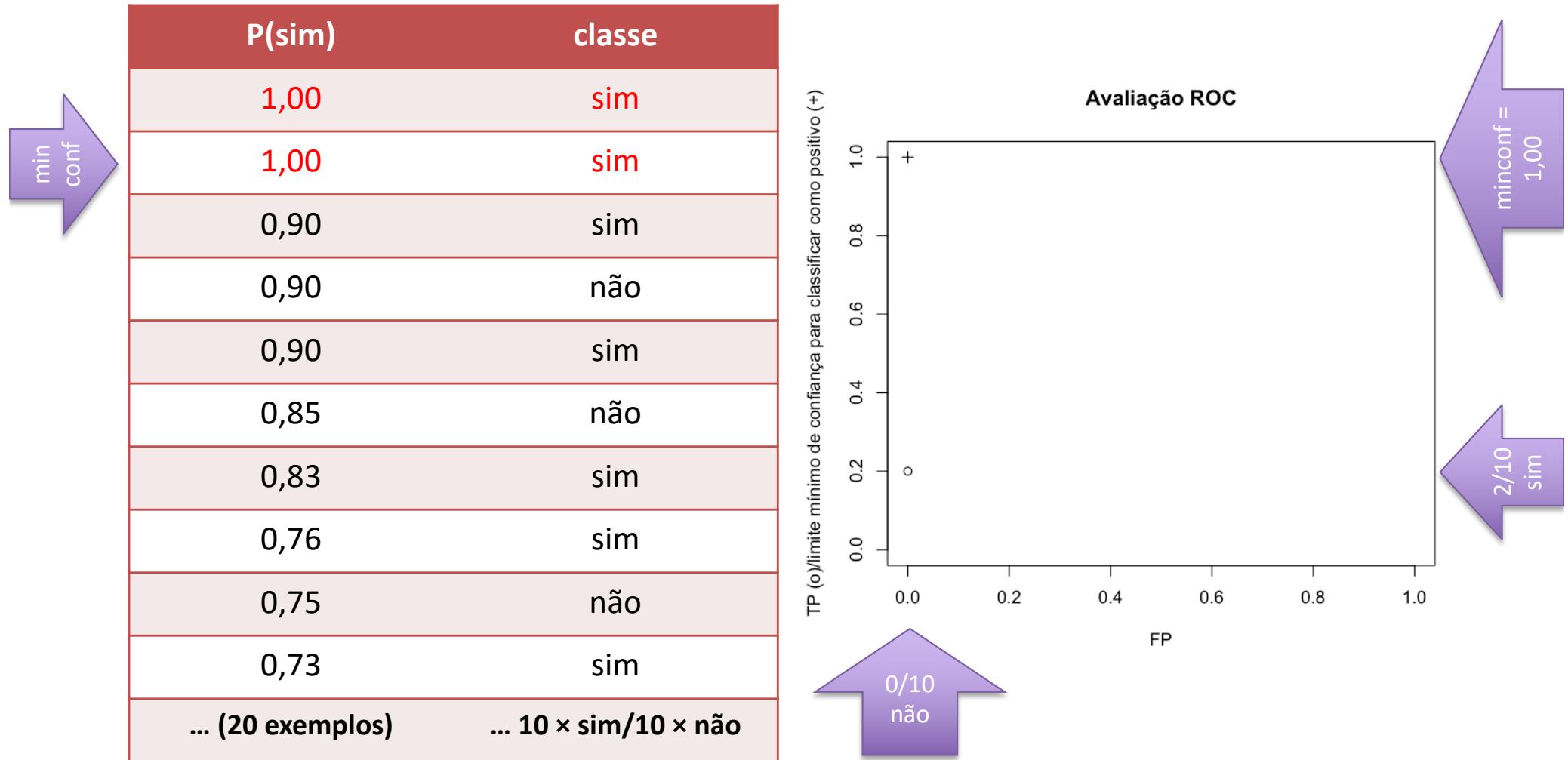


build ROC curve (1/5)

P(sim)	classe
1,00	sim
1,00	sim
0,90	sim
0,90	não
0,90	sim
0,85	não
0,83	sim
0,76	sim
0,75	não
0,73	sim
... (20 exemplos) ... 10 × sim/10 × não	



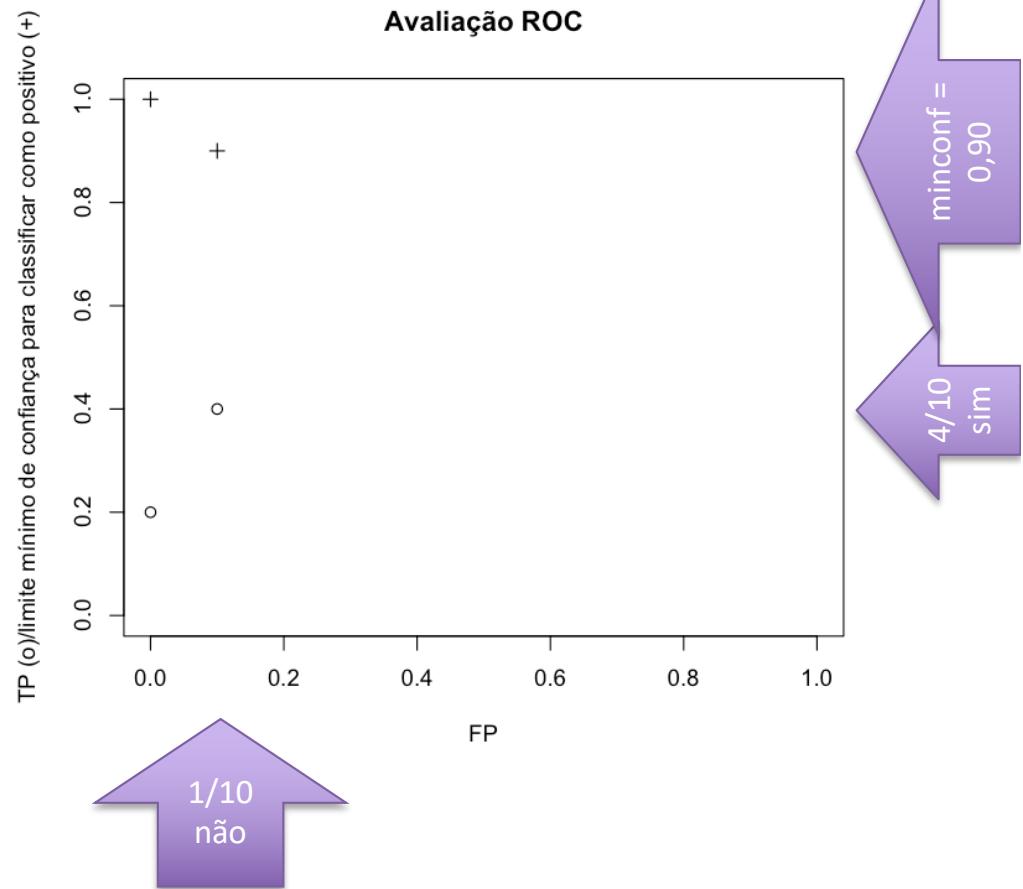
build ROC curve (2/5)



build ROC curve (3/5)

min
conf

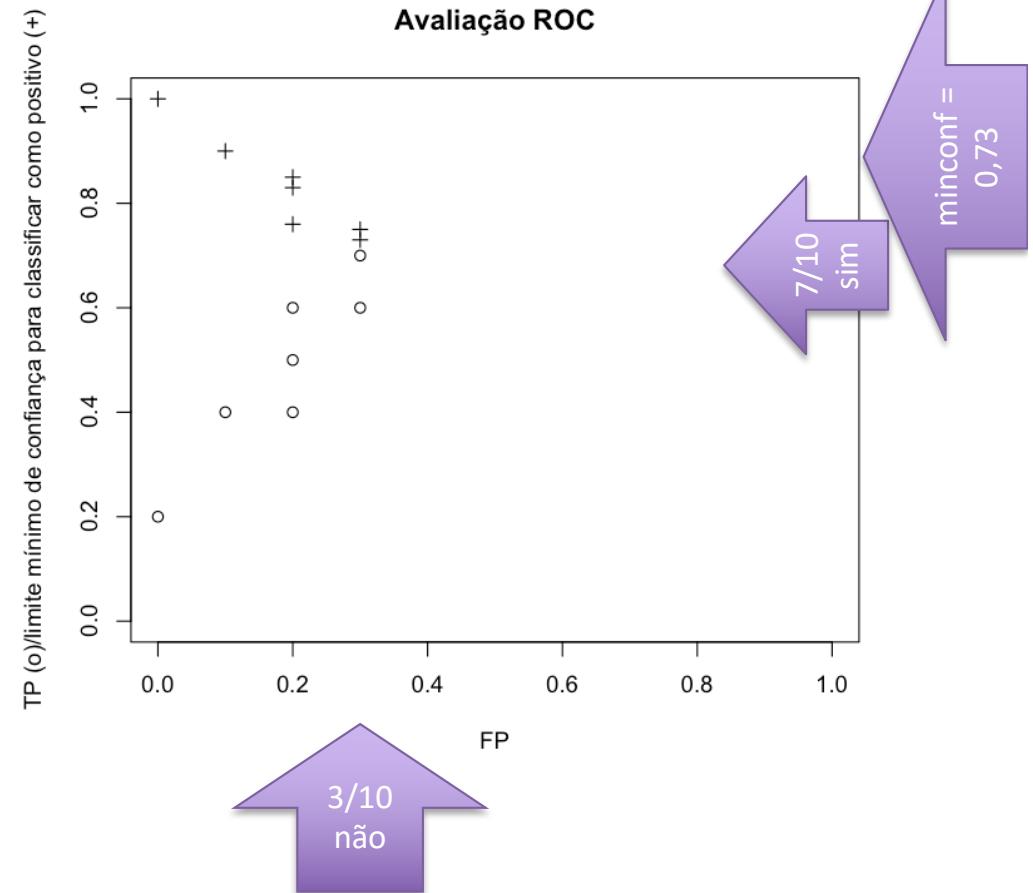
P(sim)	classe
1,00	sim
1,00	sim
0,90	sim
0,90	não
0,90	sim
0,85	não
0,83	sim
0,76	sim
0,75	não
0,73	sim
... (20 exemplos) ... 10 × sim/10 × não	



build ROC curve (4/5)

P(sim)	classe
1,00	sim
1,00	sim
0,90	sim
0,90	não
0,90	sim
0,85	não
0,83	sim
0,76	sim
0,75	não
0,73	sim
... (20 exemplos) ... 10 × sim/10 × não	

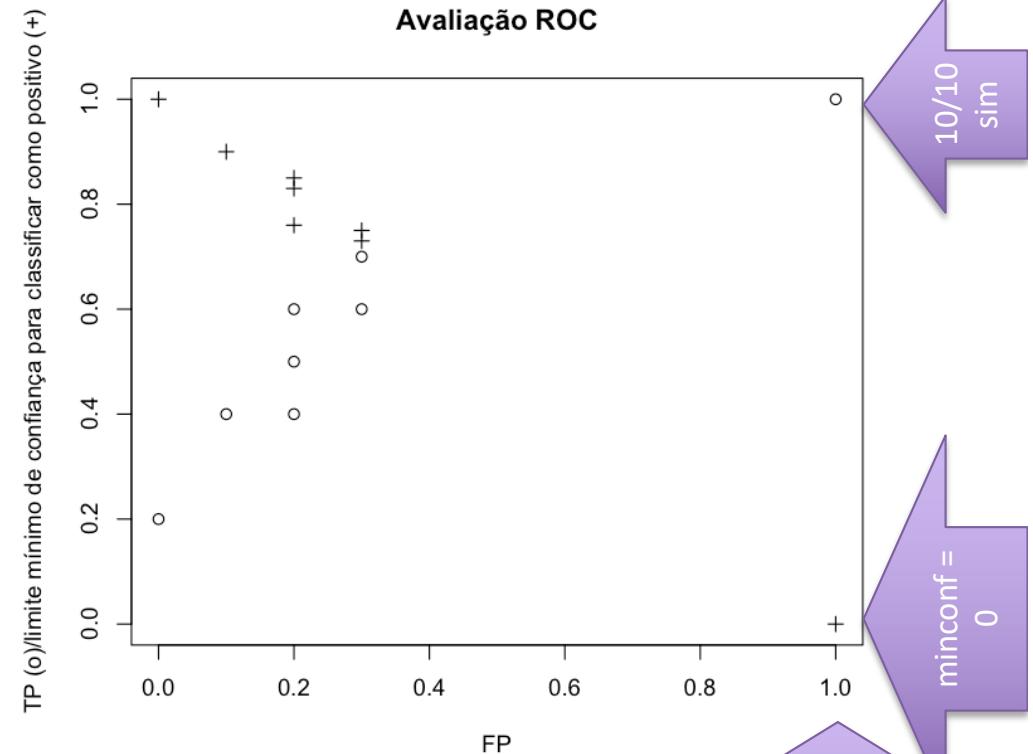
min conf →



build ROC curve (5/5)

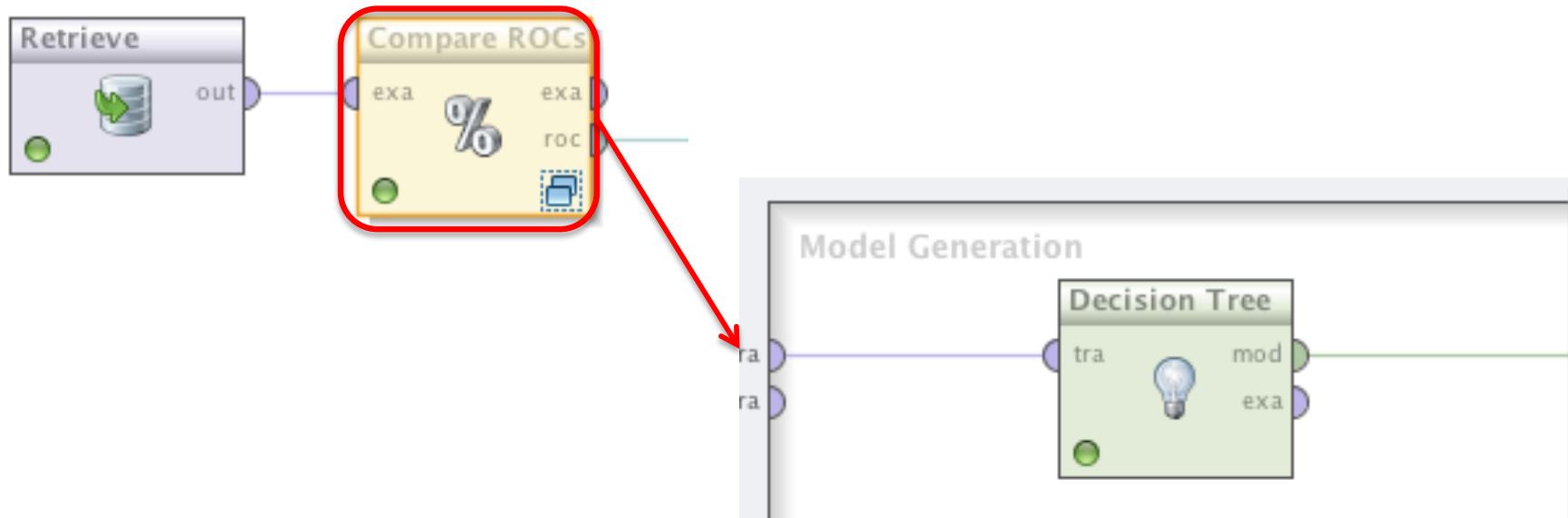
$P(\text{sim})$	classe
1,00	sim
1,00	sim
0,90	sim
0,90	não
0,90	sim
0,85	não
0,83	sim
0,76	sim
0,75	não
0,73	sim
... (20 exemplos) ... 10 × sim/10 × não	

min conf →



exercício V: sales campaign

- evaluate decision tree
 - **IMPORTANT NOTE:** first example in the training data should belong to positive class
- not very interesting
 - why?



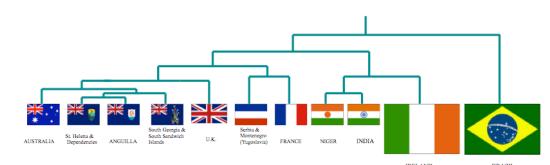
are we there yet?

- identify problems where classification is useful
- identify relevant data
- know how to analyze and use a decision tree
- know the most common evaluation measures of classification models
- understand the need to use different data for modelling and evaluation
- understand how to evaluate the results of a classification model
- superficially understand the algorithm for induction of decision trees
- understand how to use and evaluate a classification model for scoring
- address a classification problem using RapidMiner algorithms

classification (and then some...) algorithms

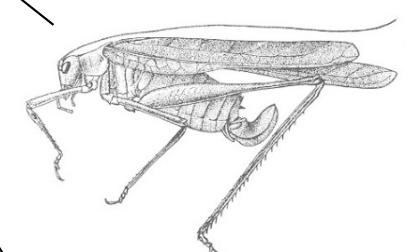
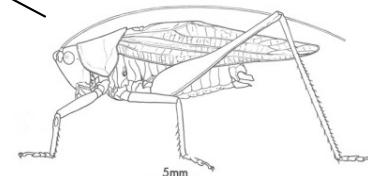
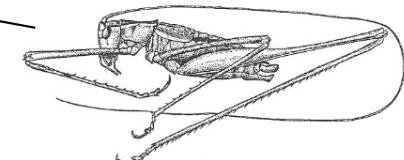
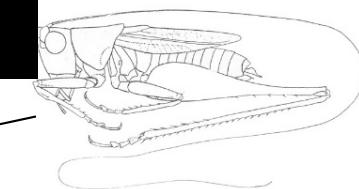
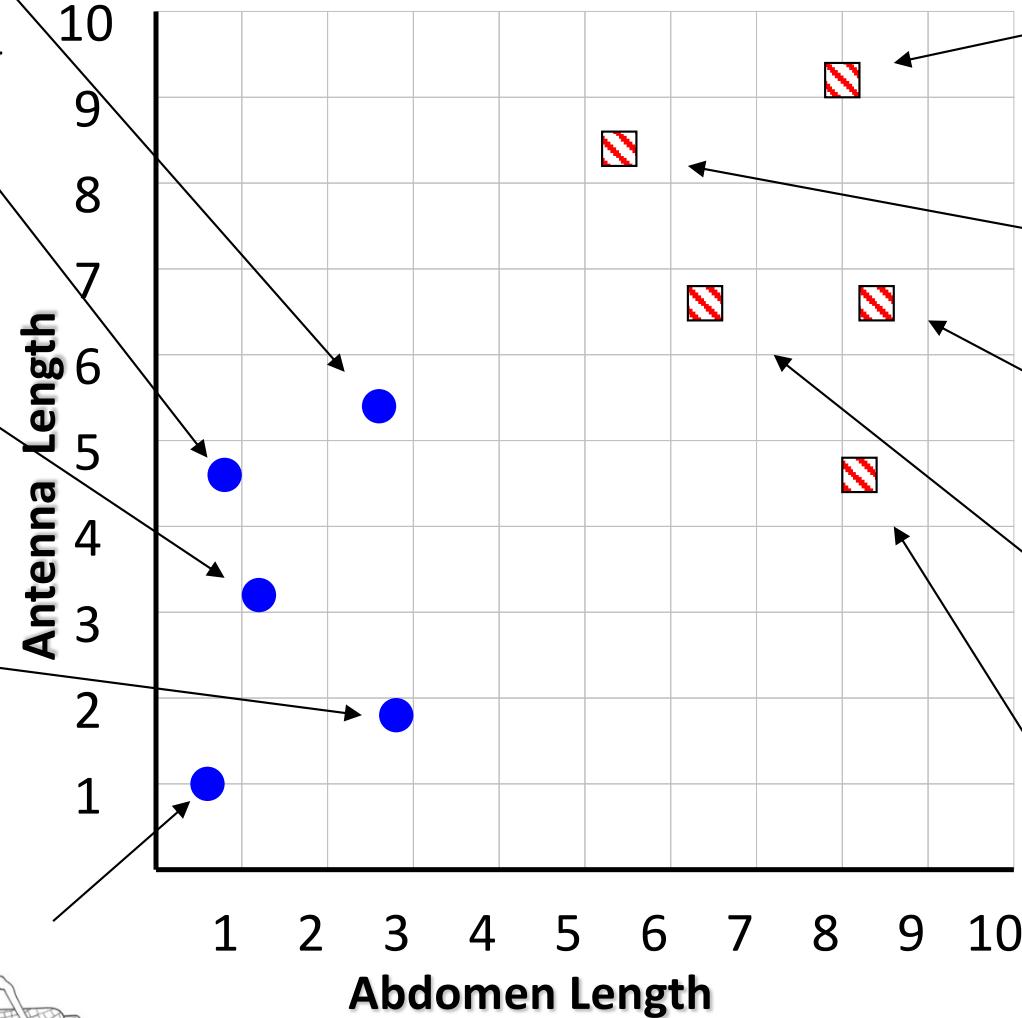
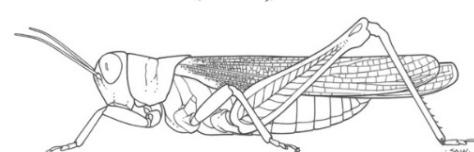
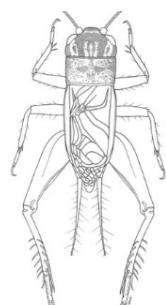
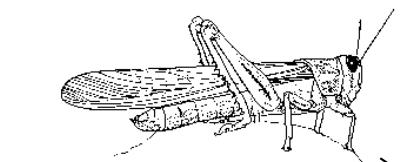
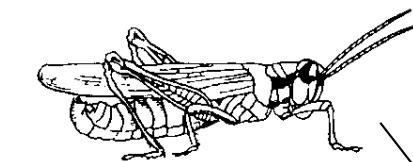
Carlos Soares

[based on materials kindly provided
by E. Keogh and J.M. Moreira]



- geometric intuition of ML algorithms
 - Linear Classifiers
 - Nearest Neighbors
 - Decision Trees
 - Naive Bayes
 - Neural Networks
 - Support Vector Machines

running (hopping?) example



Grasshoppers

carlos soares - AC

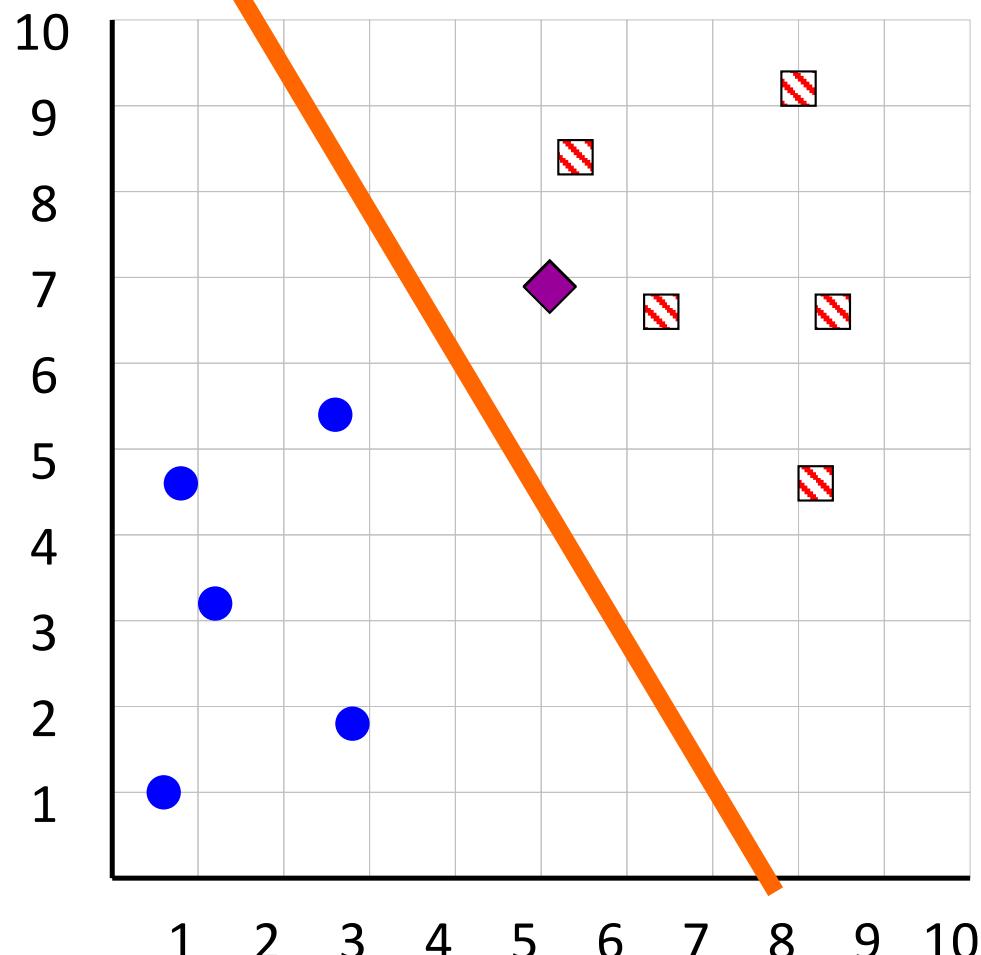
Katydids³

Simple Linear Classifier

RIA
D



R.A. Fisher
1890-1962



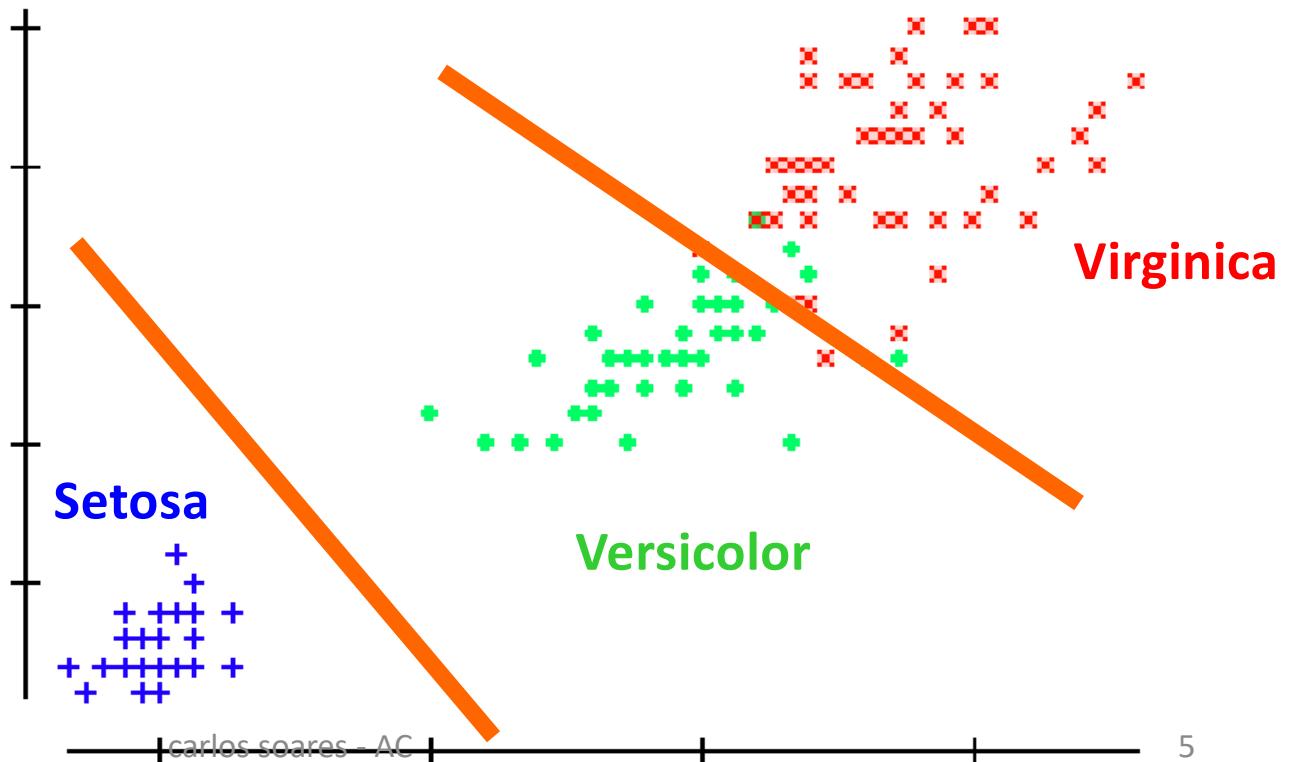
If previously unseen instance above the line
then
class is **Katydid**
else
class is **Grasshopper**

■ **Katydids**
● **Grasshoppers**

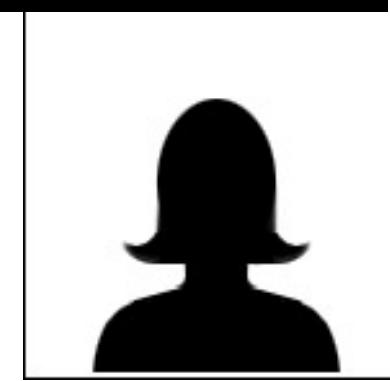
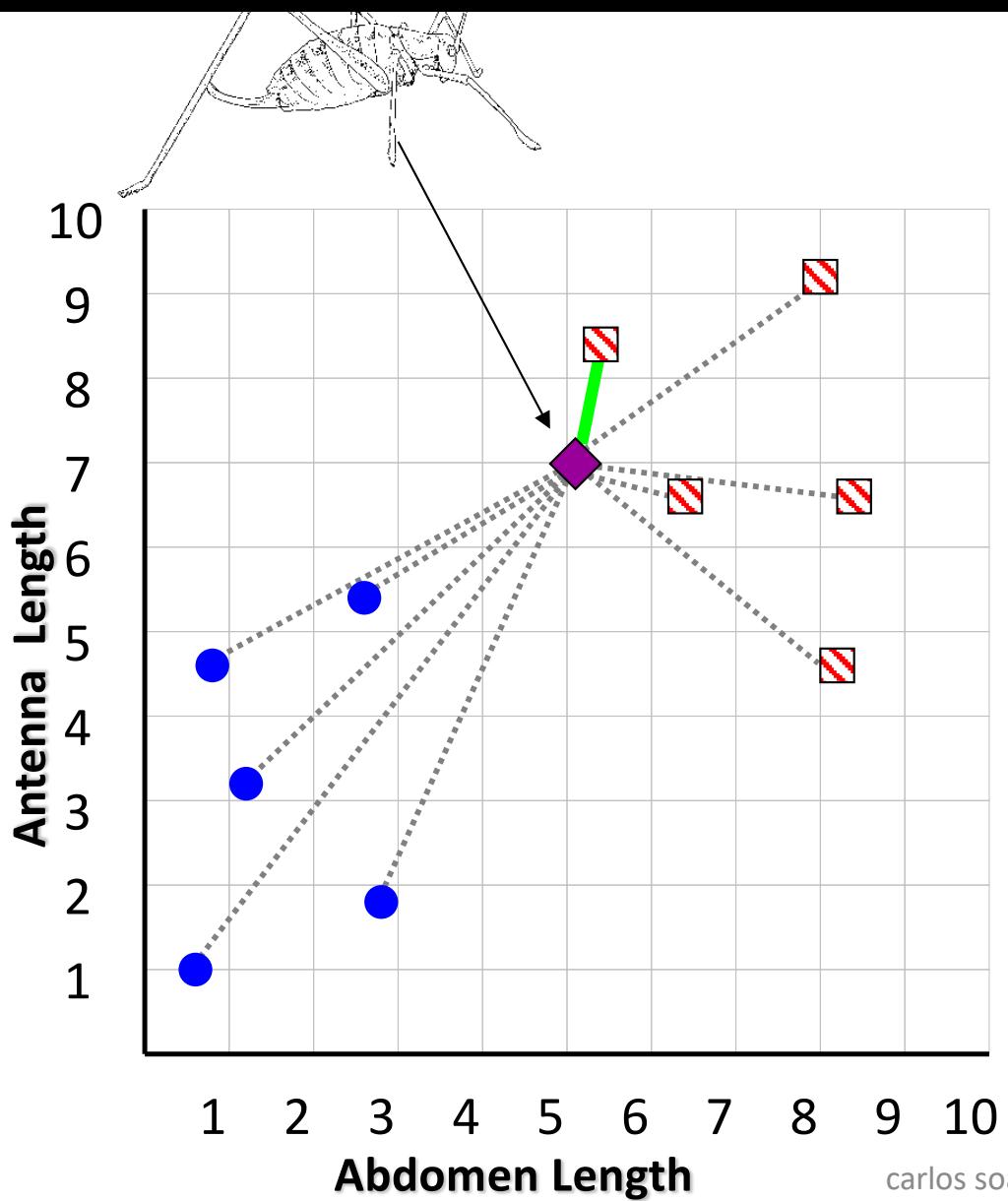
Linear Classifier for 3+ Classes

- linear classifier for N classes
- fit N-1 lines
 - Setosa vs Virginica/Versicolor
 - then
 - Virginica vs Versicolor.

If petal width > $3.272 - (0.325 * \text{petal length})$
then class = **Virginica**
Elseif petal width...



Nearest Neighbor Classifier



Evelyn Fix
1904-1965

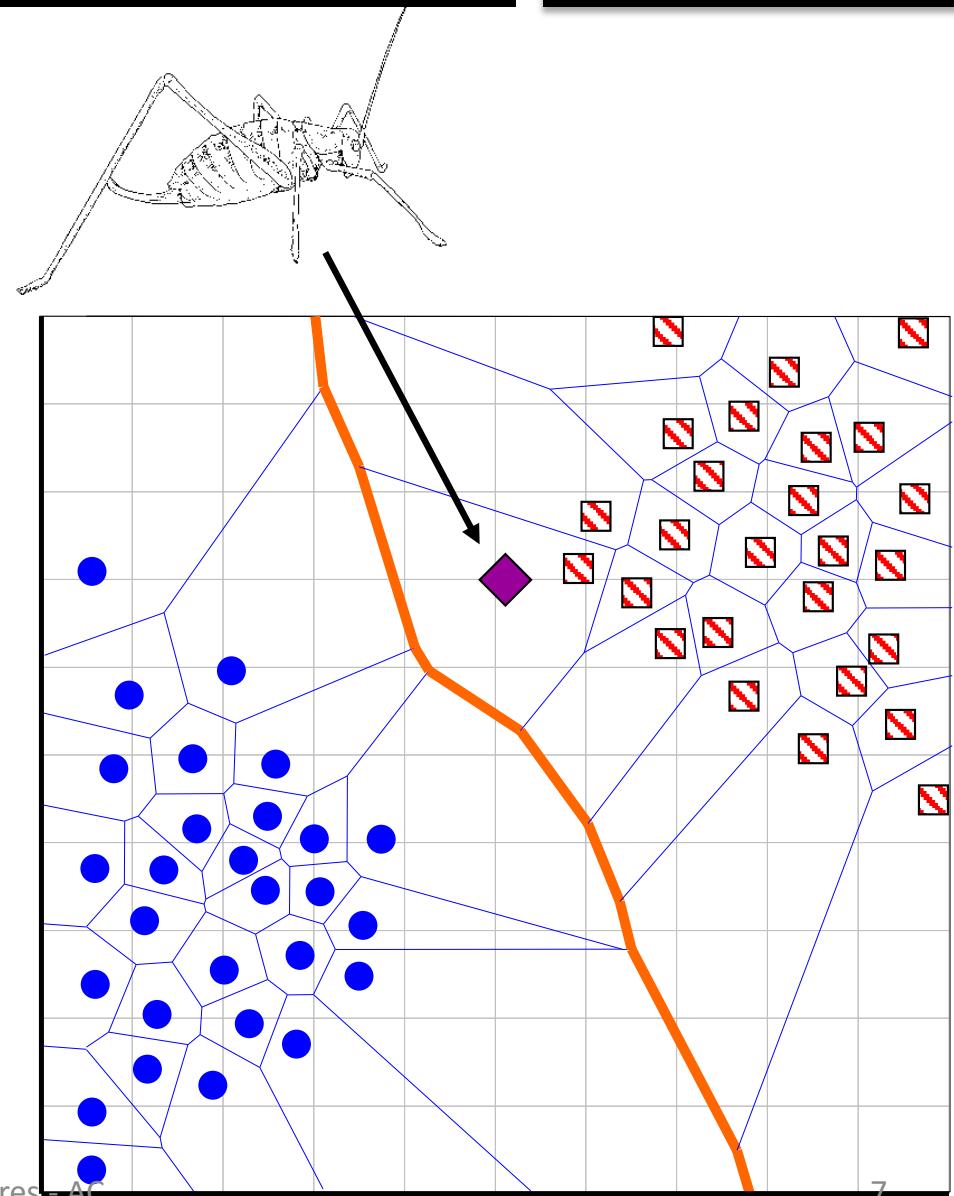
Joe Hodges
1922-2000

If the **nearest** instance to the **previously unseen instance** is a **Katydid**
class is **Katydid**
else
class is **Grasshopper**

■ **Katydid**
● **Grasshopper**

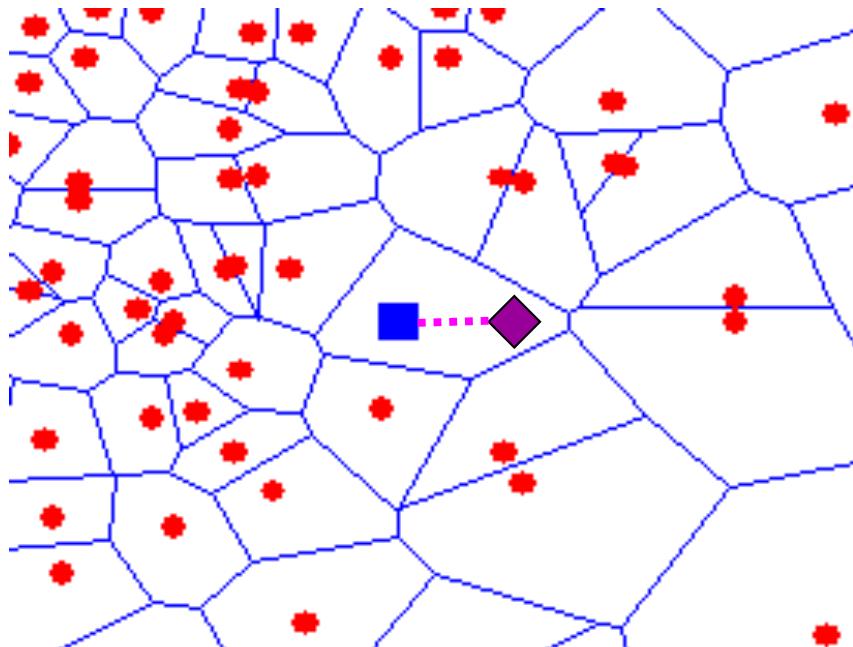
Geometrical Interpretation of Nearest Neighbor Classifier

- decision surface of a NN classifier
 - also called Dirichlet Tessellation,
 - ... Voronoi diagram
 - ... or Theissen regions
- implicit
 - never defined explicitly
 - divide the space into regions “belonging” to each instance
 - ... and corresponding class

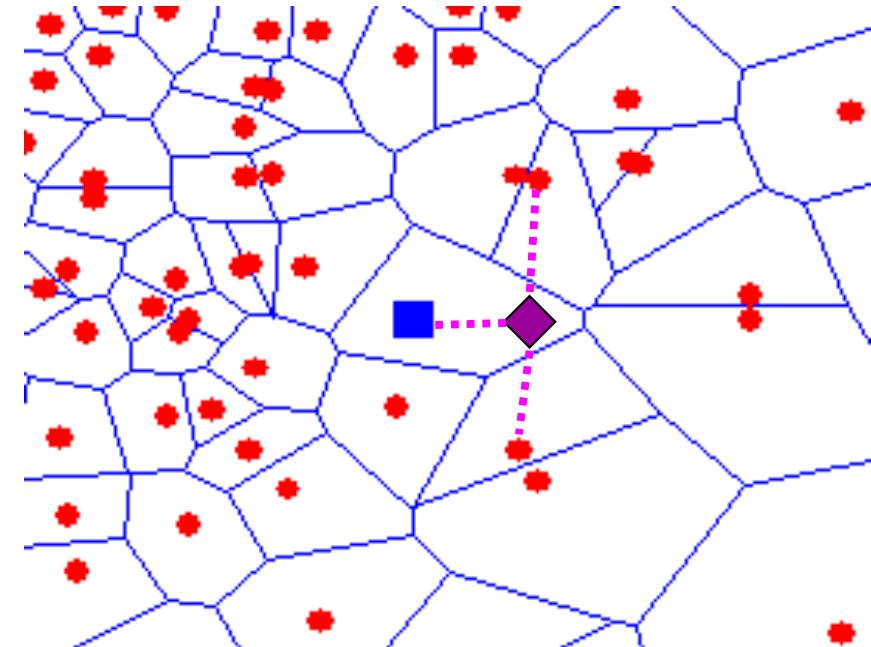


KNN Algorithm

- generalization of the nearest neighbor algorithm
 - find the nearest K instances
 - each one represents a vote
- K is typically chosen to be an odd number



$K = 1$



$K = 3$

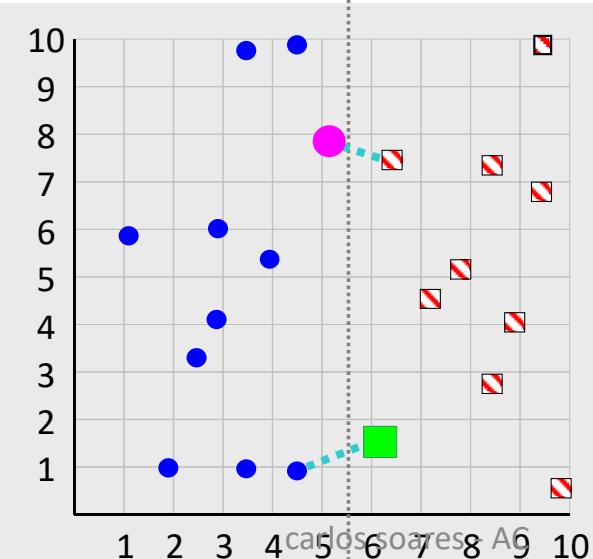
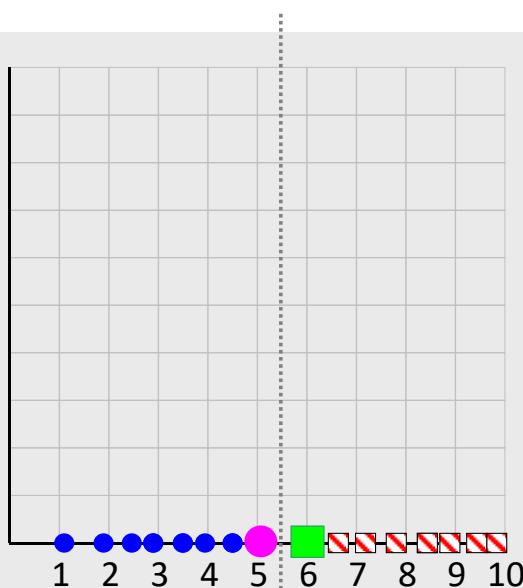
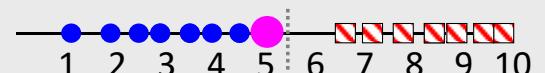
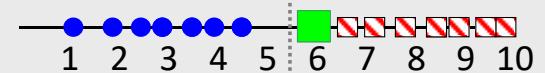
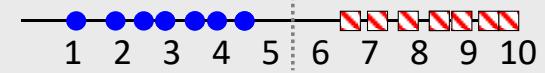
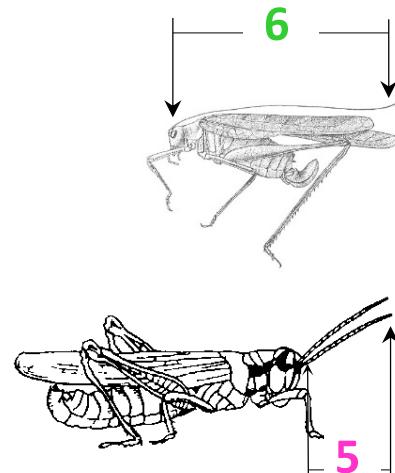
carlos soares - AC

8

Sensitivity to Irrelevant Attributes (1/2)

Suppose the following is true, if an insects antenna is longer than 5.5 it is a **Katydid**, otherwise it is a **Grasshopper**.

Using just the antenna length we get perfect classification!

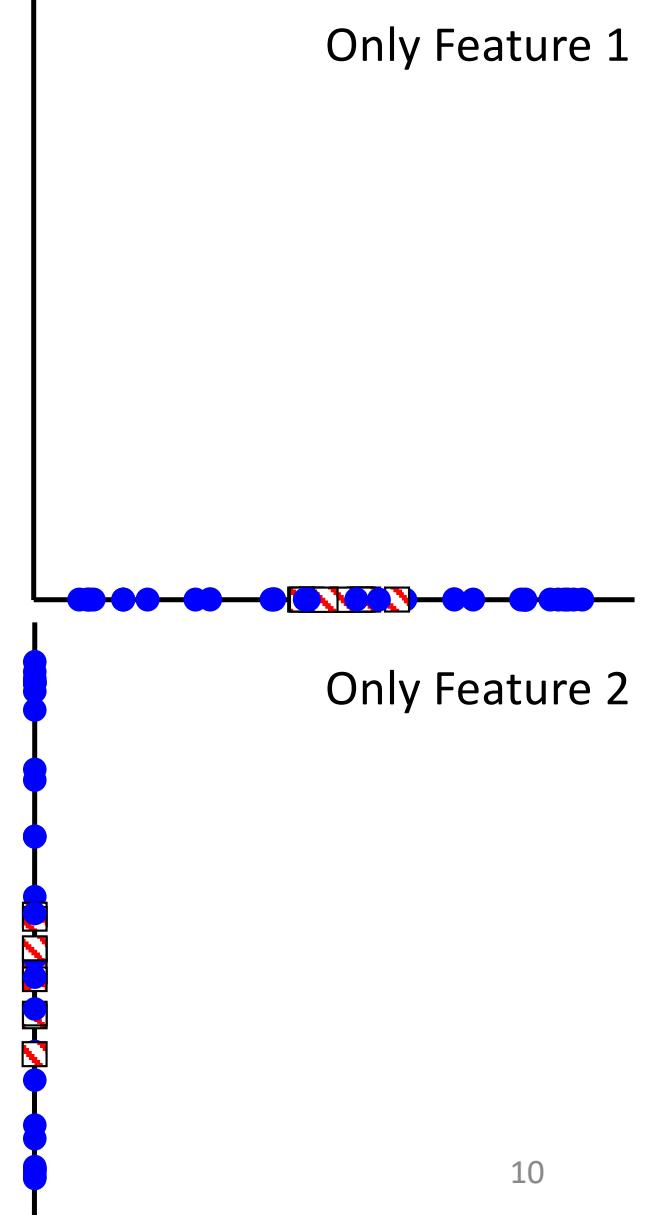
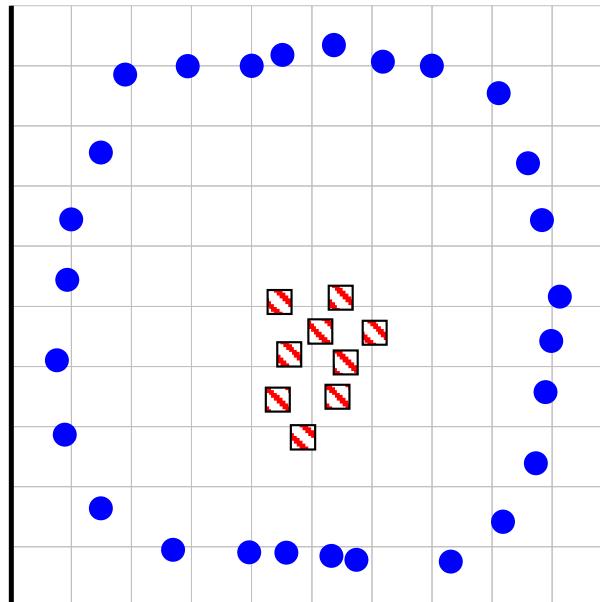


Suppose however, we add in an **irrelevant** feature, for example the insects mass.

Using both the antenna length and the insects mass with the 1-NN algorithm we get the wrong classification!

why search over feature subsets can fail

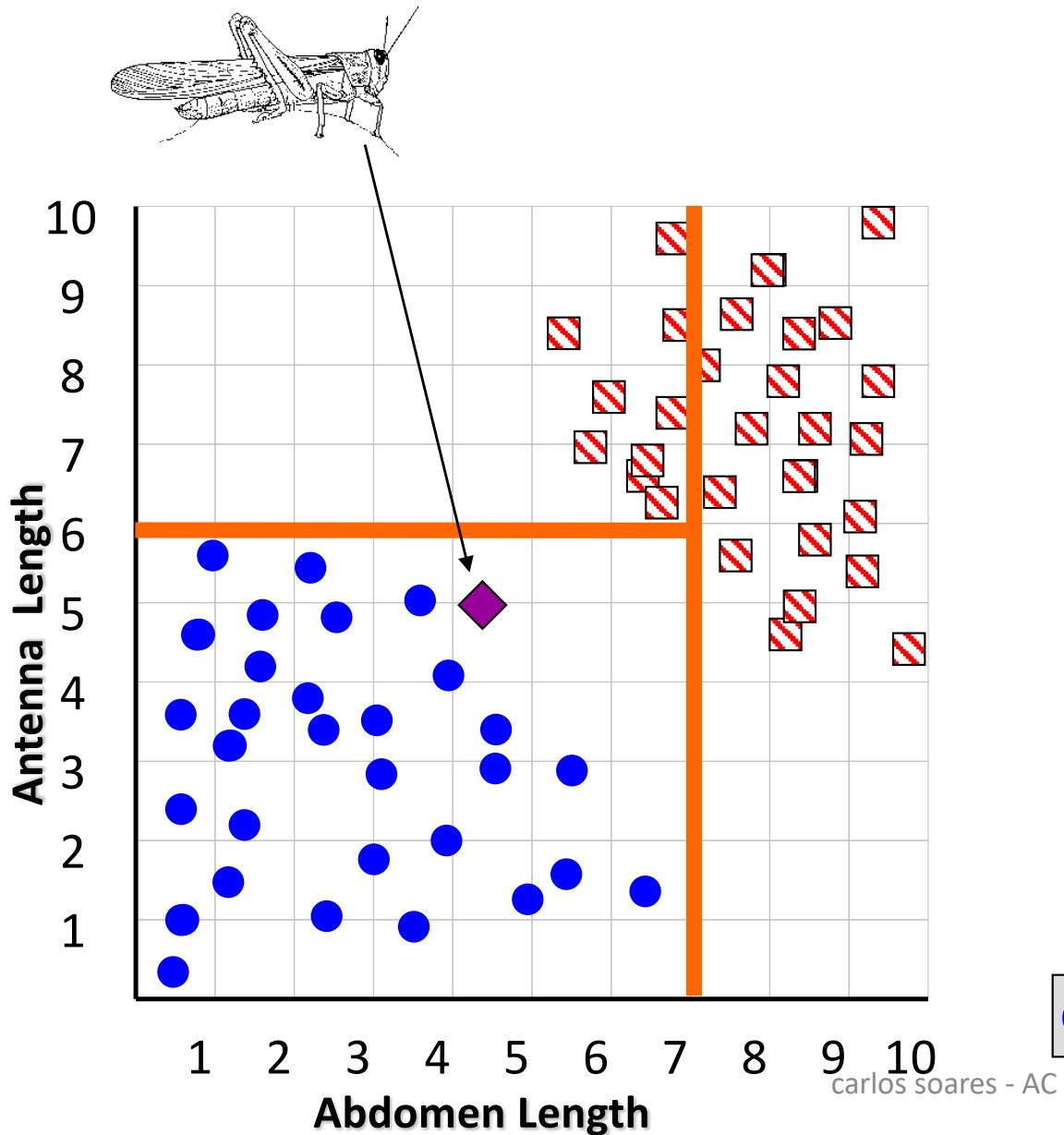
- problem
 - data with 100 features
 - Features 1 and 2 give perfect classification
 - ... remaining 98 are irrelevant
- poor results
 - 100 features
 - Feature 1 alone
 - Feature 2 alone
- only one subset gives good results
 - $2^{100} - 1$ possible subsets



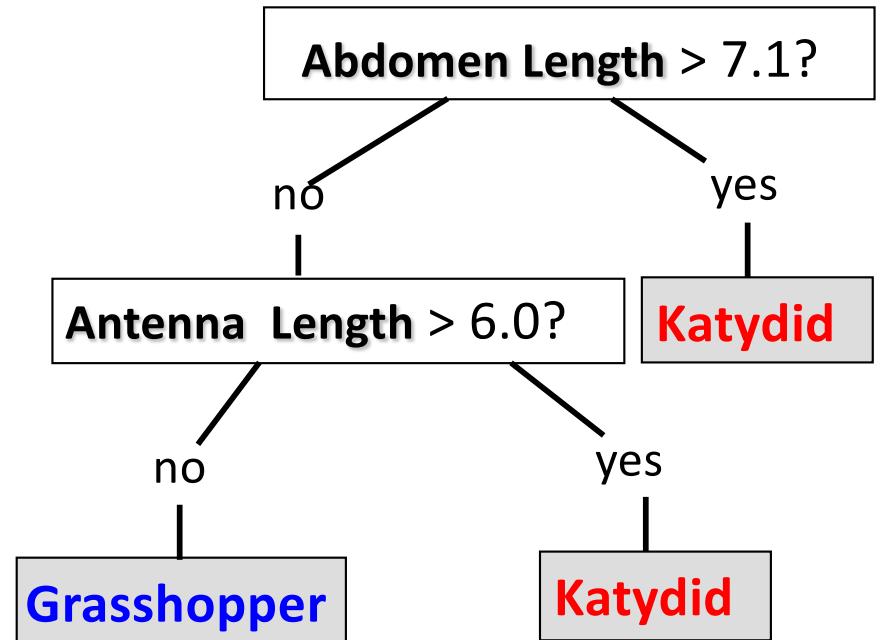
Advantages/Disadvantages of Nearest Neighbor

- Advantages
 - Simple to implement
 - Handles correlated features
 - Arbitrary class shapes
 - Defined for any distance measure
 - Handles streaming data trivially
- Disadvantages
 - Very sensitive to irrelevant features
 - Slow classification time for large datasets
 - Works best for real valued datasets

Decision Tree Classifier



Ross Quinlan



Avoid Overfitting in Classification

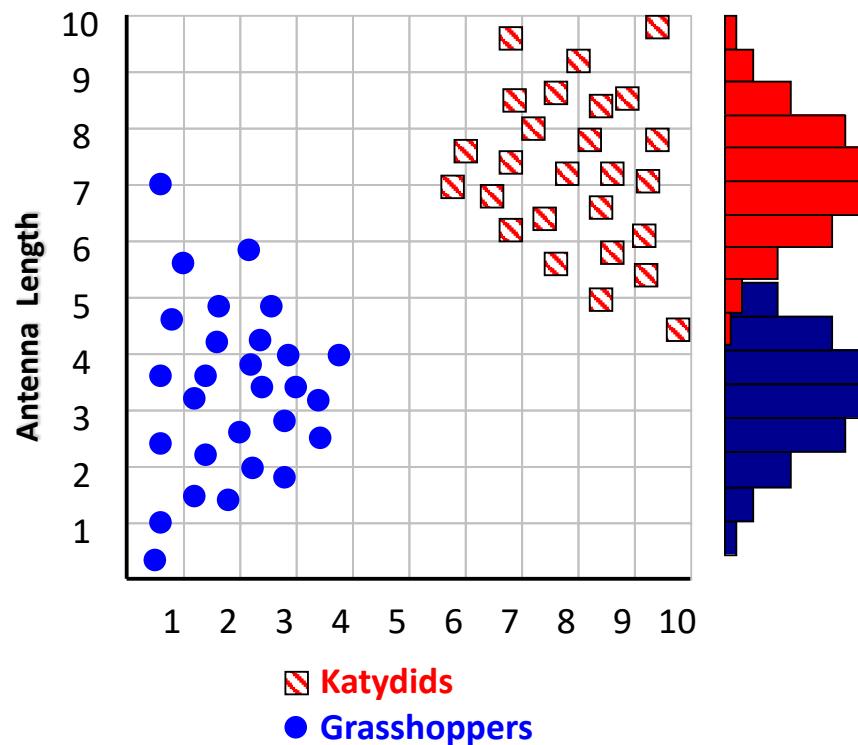
- Overfitting the training data
 - Too many branches
 - some may reflect anomalies due to noise or outliers
 - Poor accuracy for unseen samples
- Avoid overfitting
 - Prepruning
 - Halt tree construction early
 - do not split a node if this would result in the goodness measure falling below a threshold
 - ... difficult to choose an appropriate threshold
 - Postpruning
 - Get a sequence of progressively pruned trees
 - removing branches from a “fully grown” tree
 - Use a set of data different from the training data to decide which is the “best pruned tree”

Advantages/Disadvantages of Decision Trees

- Advantages
 - Easy to understand
 - Easy to generate rules
- Disadvantages
 - Overfitting
 - Does not handle correlated features very well
 - classifies by rectangular partitioning
 - Can be quite large
 - pruning is necessary

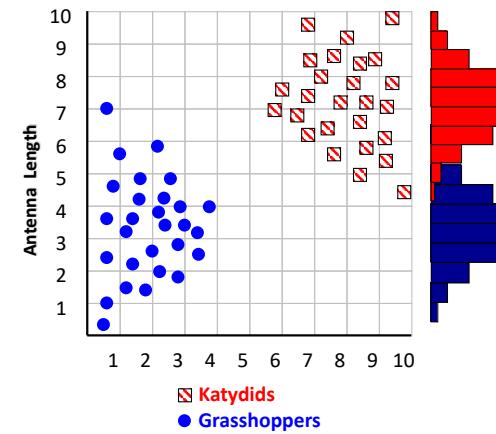
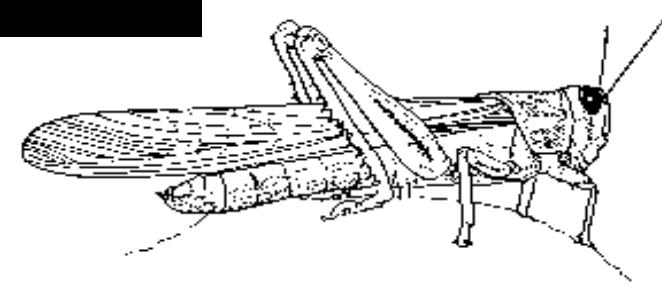
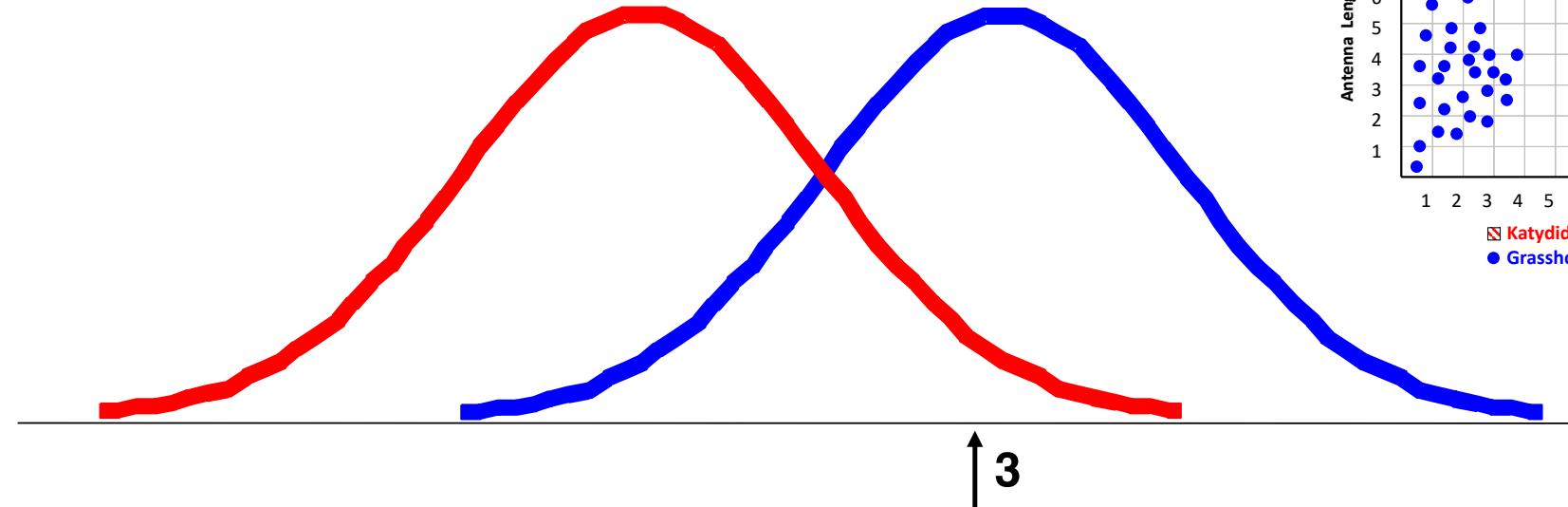
the distribution of bugs

- class histogram
 - the more data available
 - ... the more reliable



... can be used for prediction

- insect with 3 units long antennae
- ... more *probably* a **Grasshopper** or a **Katydid**?
 - given the distributions of antennae lengths we have seen



$p(c_j | d)$ = probability of class c_j , given that we have observed d

... but is it reliable?

how much evidence supports
 $p(c_j | d)$?

Name	Over 170cm	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

Bayes classifies

- Bayesian classifiers use **Bayes theorem**, which says

$$p(c_j | d) = \frac{p(d | c_j) p(c_j)}{p(d)}$$

– $p(c_j | d)$ = probability of instance d being in class c_j ,

This is what we are trying to compute

– $p(d | c_j)$ = probability of generating instance d given class c_j ,

We can imagine that being in class c_j , causes you to have feature d with some probability

– $p(c_j)$ = probability of occurrence of class c_j ,

This is just how frequent the class c_j , is in our database

– $p(d)$ = probability of instance d occurring

This can actually be ignored, since it is the same for all classes

... assuming independence

if attributes have independent distributions

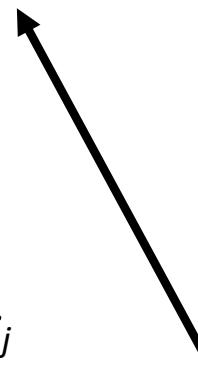
$$p(d|c_j) = p(d_1|c_j) * p(d_2|c_j) * \dots * p(d_n|c_j)$$



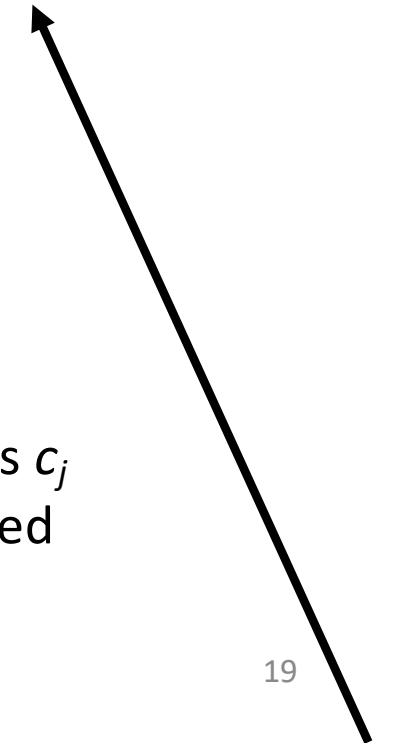
The probability of class c_j generating instance d , equals....



The probability of class c_j generating the observed value for feature 1, multiplied by..



The probability of class c_j generating the observed value for feature 2, multiplied by..

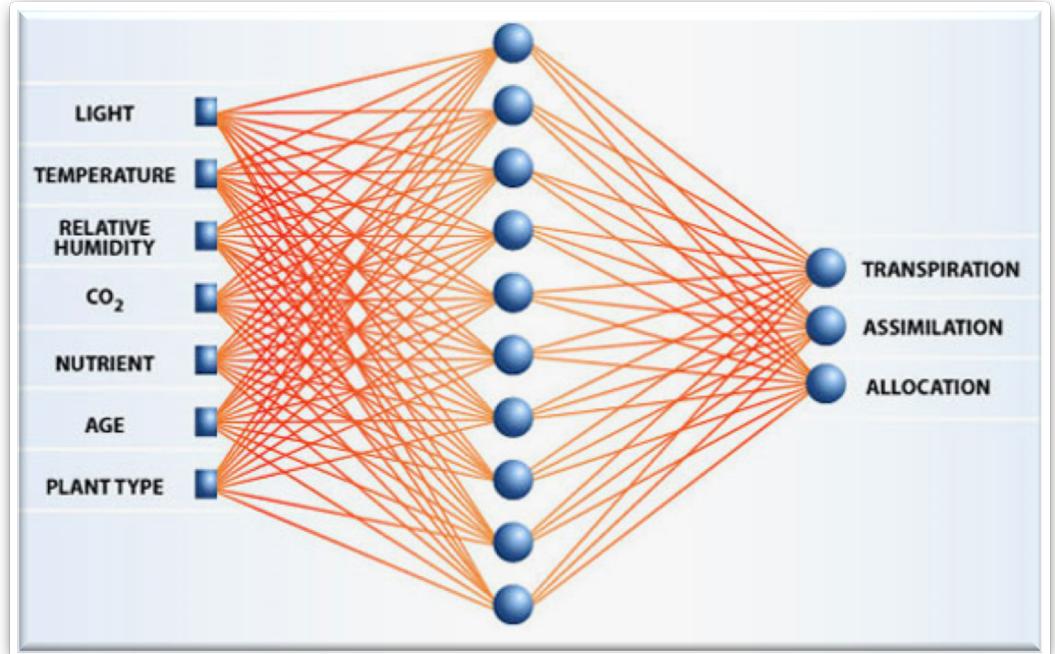


Advantages/Disadvantages of Naive Bayes

- Advantages
 - Fast
 - to train
 - single scan
 - to classify
 - Not sensitive to irrelevant features
 - Handles real and discrete data
 - Handles streaming data well
- Disadvantages
 - Assumes independence of features

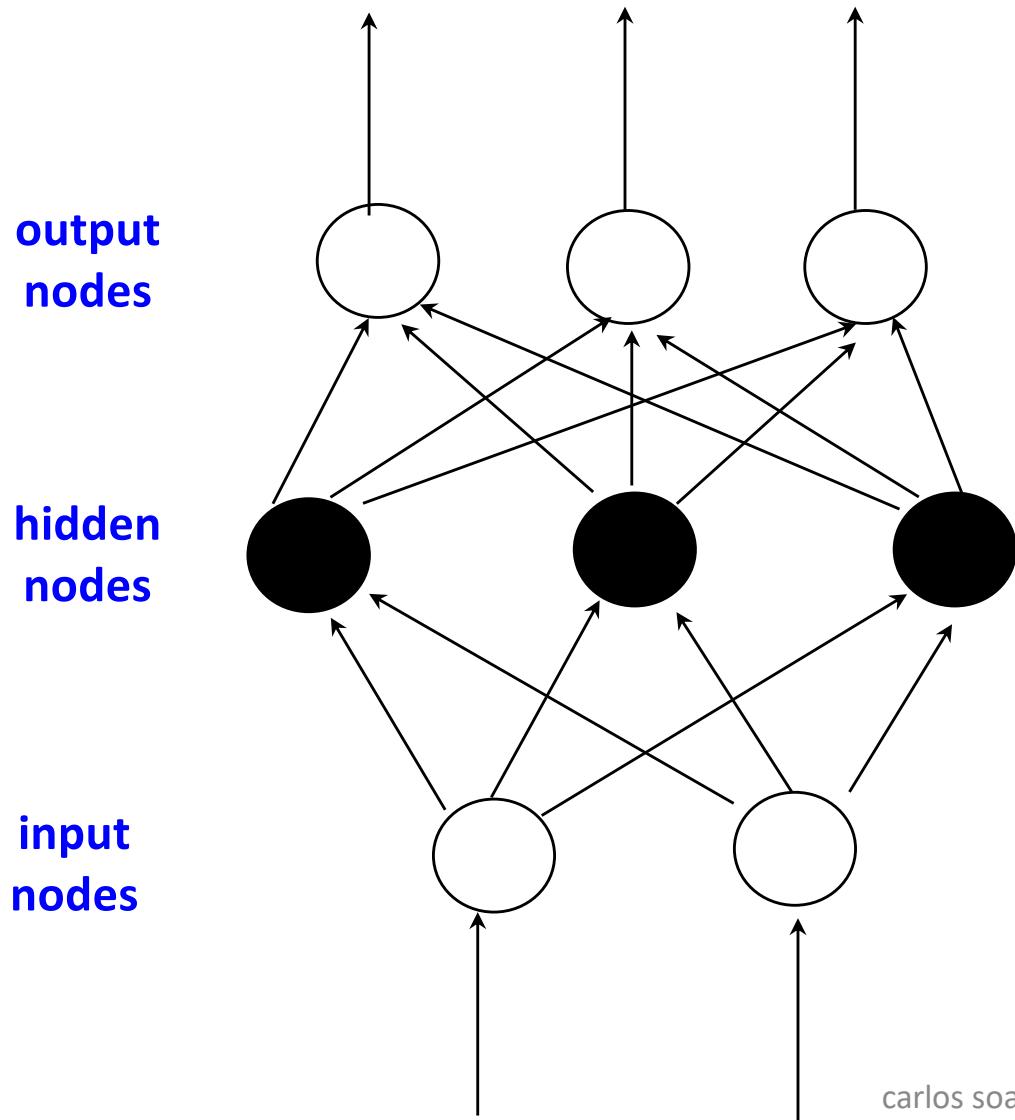
neural network learning

- set of neurons
 - input/output units
- connected
- ... with weights
- (supervised) learning
 - adjust weights to ensure outputs to given inputs are the expected ones
- predicting
 - feed input values
 - collect outputs



http://aemc.jpl.nasa.gov/activities/bio_regen.cfm

summary: feedforward



$$\hat{y}_k = \frac{1}{1 + e^{-w_{0k} + \sum_{i \in \{\text{all units feeding into unit } k\}} w_{ik} y_i}}$$

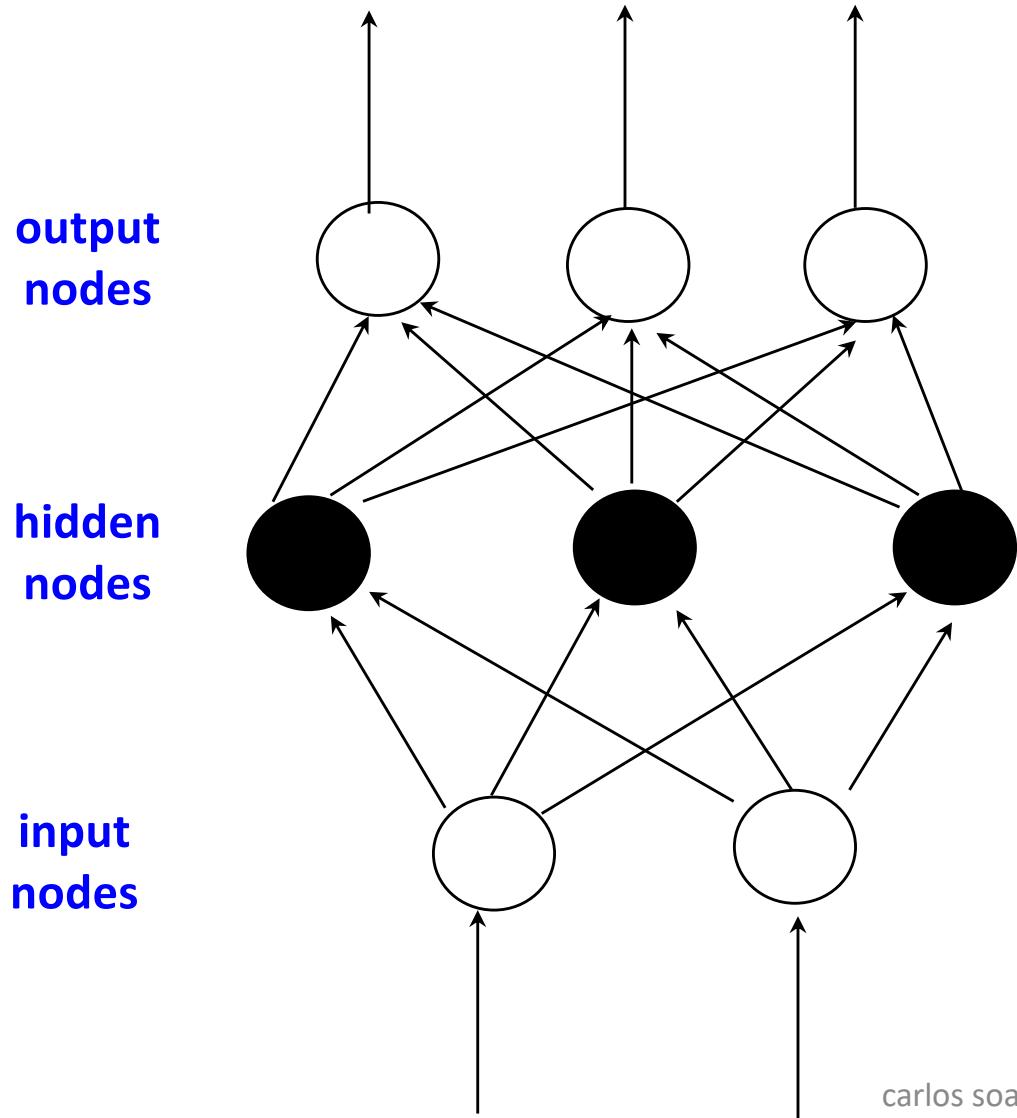
$w_{j,k}$ weights

$$y_i = \frac{1}{1 + e^{-w_{0i} + \sum_{j \in \{\text{all units feeding into unit } i\}} w_{ji} x_j}}$$

$w_{i,j}$ weights

x_i input values

summary: backprop



$$\delta_k = \hat{y}_k(1 - \hat{y}_k)(\hat{y}_k - y_k)$$

$$w_{j,k}^{new} = w_{j,k} + r \cdot y_j \cdot \delta_k$$

$$\delta_j = y_j(1 - y_j) \sum_{k \in \{\text{all units fed by this unit}\}} \delta_k$$

$$w_{i,j}^{new} = w_{i,j} + r \cdot x_i \cdot \delta_j$$

discussion



- universal
 - fit any continuous function
- versatile
 - output may be one or more discrete and real values
- online
 - application and learning are intertwined
- robust
 - errors and noisy data
- fast
 - ... application to new examples
- parallel



- slow
 - ... training
- low usability
 - empirical parameter tuning
 - network topology and learning rate
- low interpretability
 - understand the weights
- low adaptability
 - not easy to incorporate domain knowledge

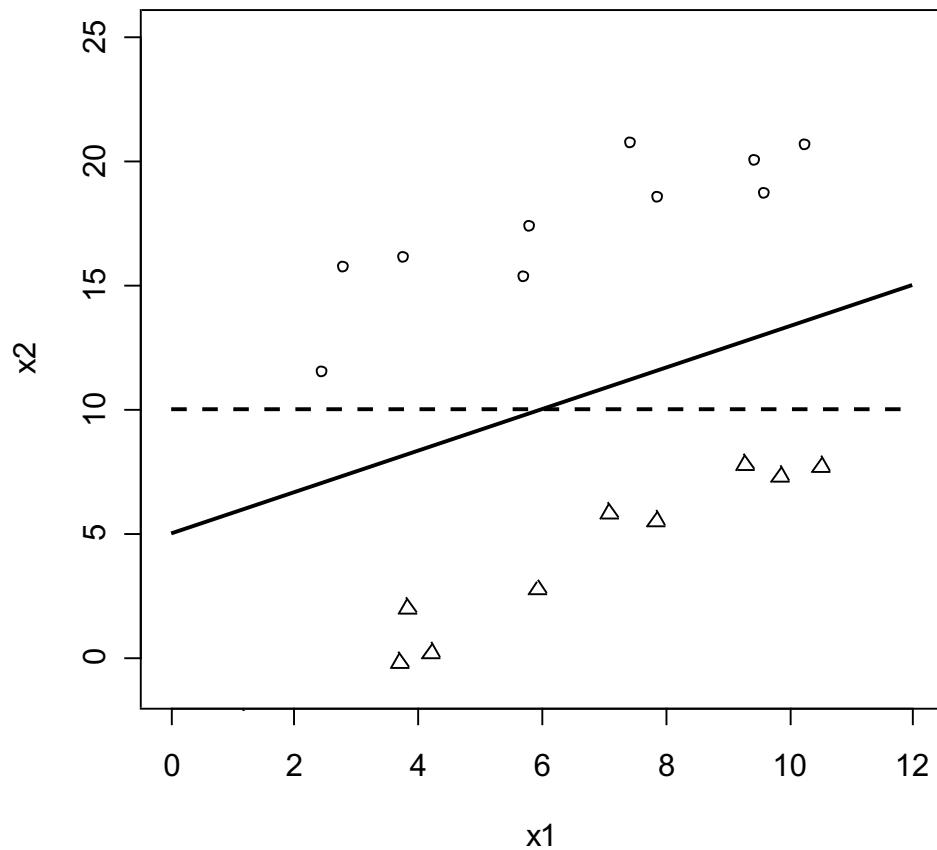
universal approximator: the whole story

- mlps are a class of universal approximators
 - 1 hidden layer
- so what's the catch?
 - provided sufficiently many hidden units...

overview

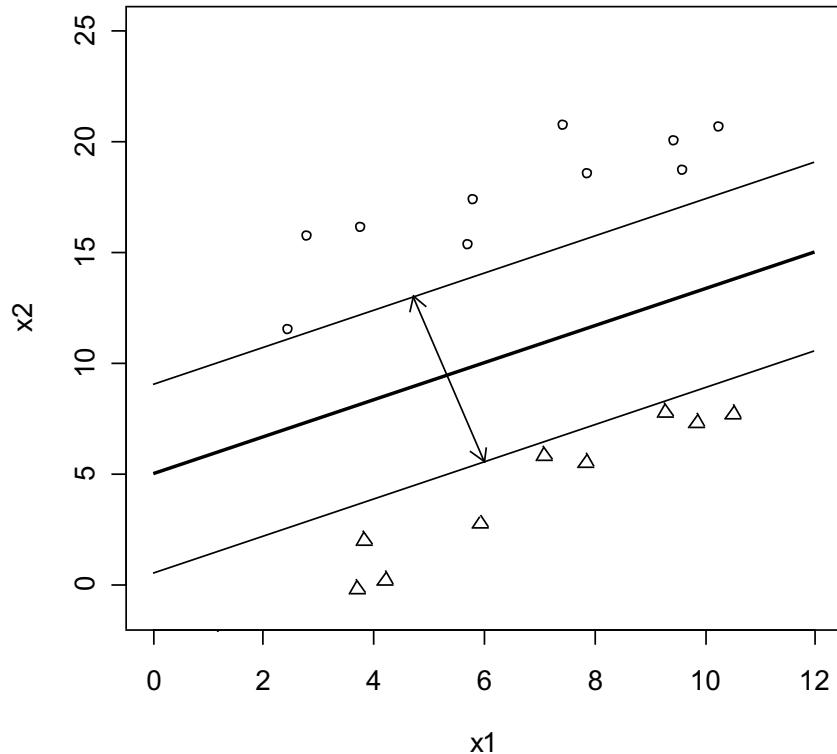
- Support Vector Machines (SVM)
 - linear learning machines with maximization of margin
 - **better separation between classes**
 - duality
 - **higher robustness to the curse of dimensionality**
 - kernel trick
 - **non-linear models**
- according to Bennet & Campbell, “Support Vector Machines: Hype of Hallelujah?”, SIGKDD Explorations, 2000
 - geometric intuition
 - elegant math
 - theoretical guarantees
 - practical algorithms

best separating hyperplane?



margin maximization: intuition

- margin, γ :
 - sum of the shortest distances between points of each class and the separating hyperplane
- margin maximization
 - hyperplane “farthest away” from both classes simultaneously
- less risk of overfitting!



margin maximization: formal definition

- it can be proved that

$$\gamma = \frac{1}{\|\mathbf{w}\|_2}$$

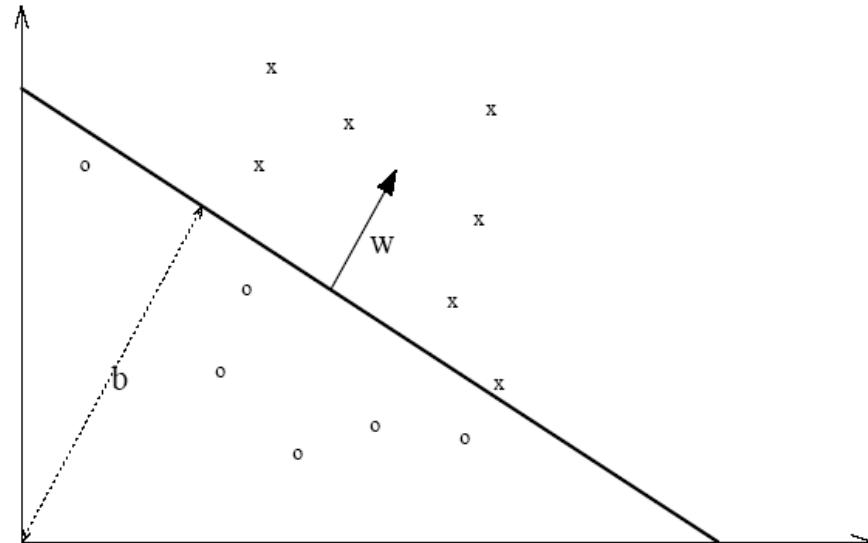
- ... thus, maximizing the margin is the same as

minimize

$$\|\mathbf{w}\|_2$$

subject to $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1$
the constraints $i = 1, \dots, l$

- quadratic programming problem
 - i.e. a lot of work in the area of optimization in this problem can be reused, most importantly...



fonte: www.kernel-machines.org

... duality

- dual problem

$$\text{maximize} \quad \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

subject to
the constraints

$$\begin{aligned} \sum_{i=1}^l y_i \alpha_i &= 0 \\ \alpha_i &\geq 0, i = 1, \dots, l \end{aligned}$$

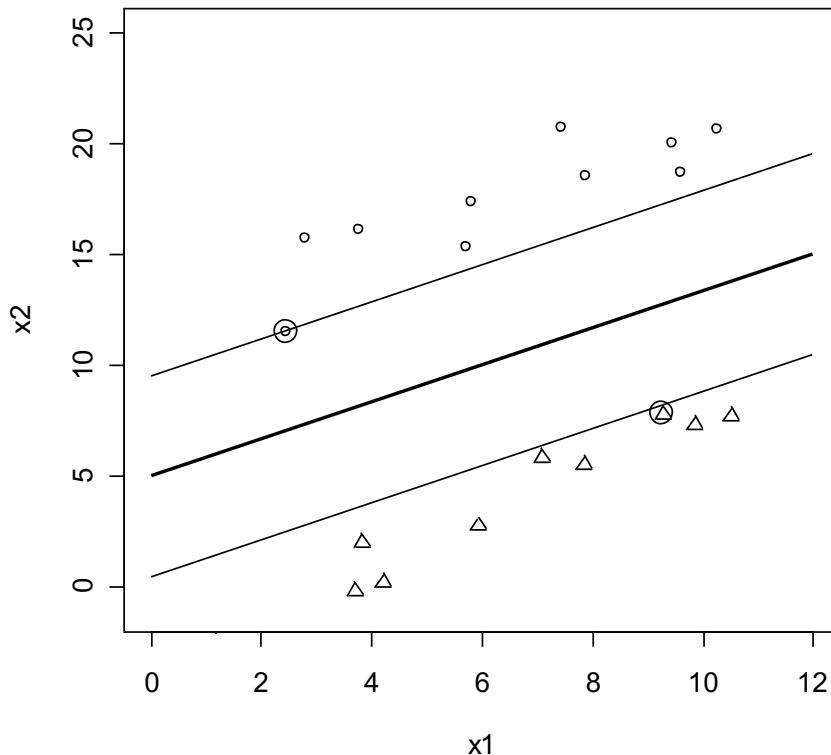
- linear prediction problem

$$f(\mathbf{x}_{new}) = \sum_{i=1}^l \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x}_{new} \rangle + b$$

model

- subset of examples that determine the margin
 - frontier
 - other points are irrelevant

$$f(\mathbf{x}_{new}) = \sum_{i=1}^l \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x}_{new} \rangle + b$$



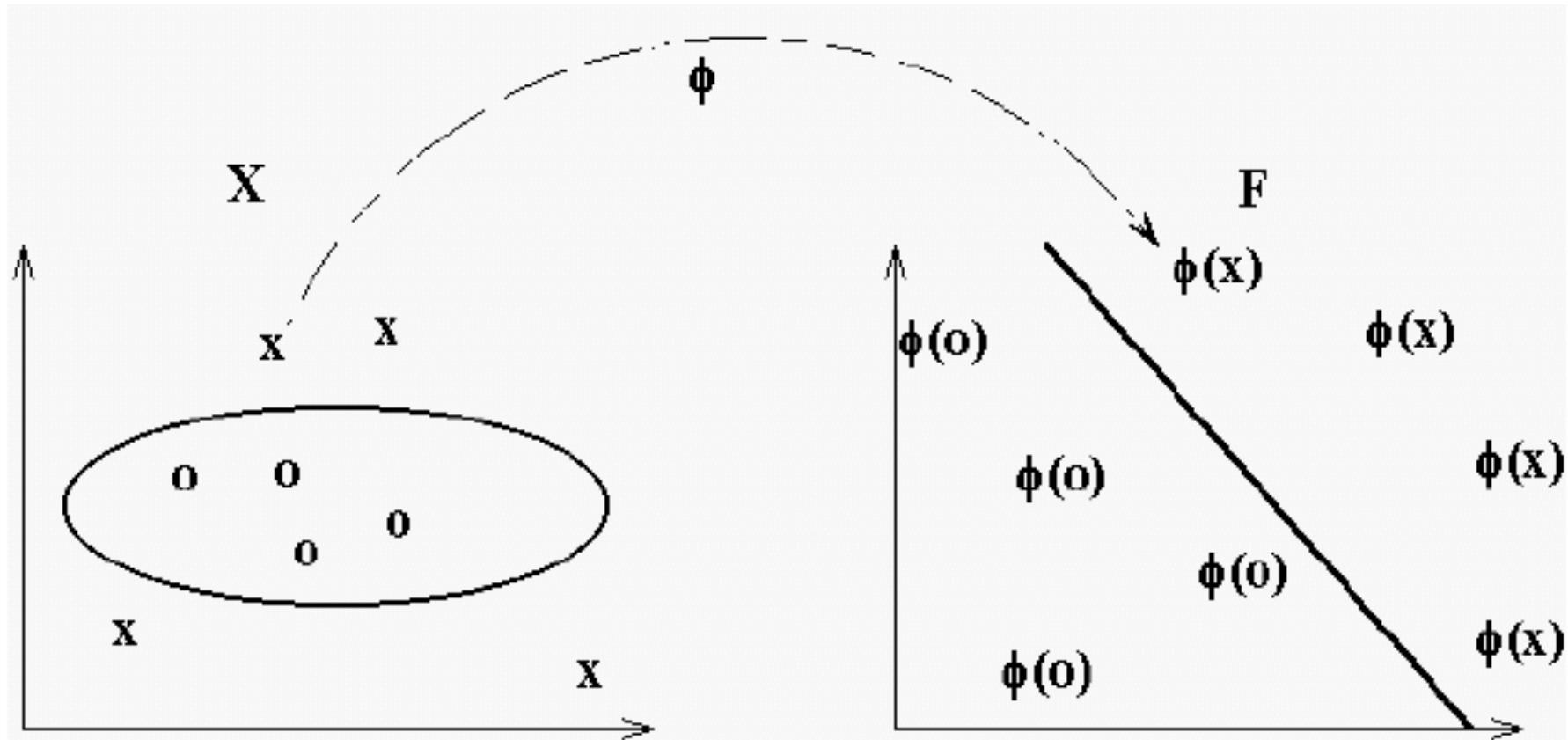
pros

- QP methods are very mature
- statistical learning theory
 - bounds to the generalization error based on the training error
- results independent of initial conditions
 - order of presentation of examples
 - initializations
- convex problem
 - no local minima
 - ... reducing the probability of overfitting
- dual is independent of number of attributes
 - minimizing effect of the curse of dimensionality

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

non-linear functions

- map attributes to space where linear discrimination is possible



fonte: www.kernel-machines.org

but, which space?

- “A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in low-dimensional space”
 - Cover (95)
- i.e., the bigger, the better... or maybe not
 - theoretical challenge: curse of dimensionality
 - practical challenge: computational resources
 - e.g. memory

dual problem in space...

- dual problem

$$\text{maximize} \quad \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle \phi(x_i) \cdot \phi(x_j) \rangle$$

subject to
the constraints

$$\begin{aligned} \sum_{i=1}^l y_i \alpha_i &= 0 \\ \alpha_i &\geq 0, i = 1, \dots, l \end{aligned}$$

- (non-)linear prediction function

$$f(\mathbf{x}_{new}) = \sum_{i=1}^l \alpha_i y_i \langle \phi(x_i) \cdot \phi(\mathbf{x}_{new}) \rangle + b$$

the kernel trick

- kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle$
- ... integrated in the previous method

maximize $\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$

subject to
the constraints $\sum_{i=1}^l y_i \alpha_i = 0$
 $\alpha_i \geq 0, i = 1, \dots, l$

$$f(\mathbf{x}_{new}) = \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_{new}) + b$$

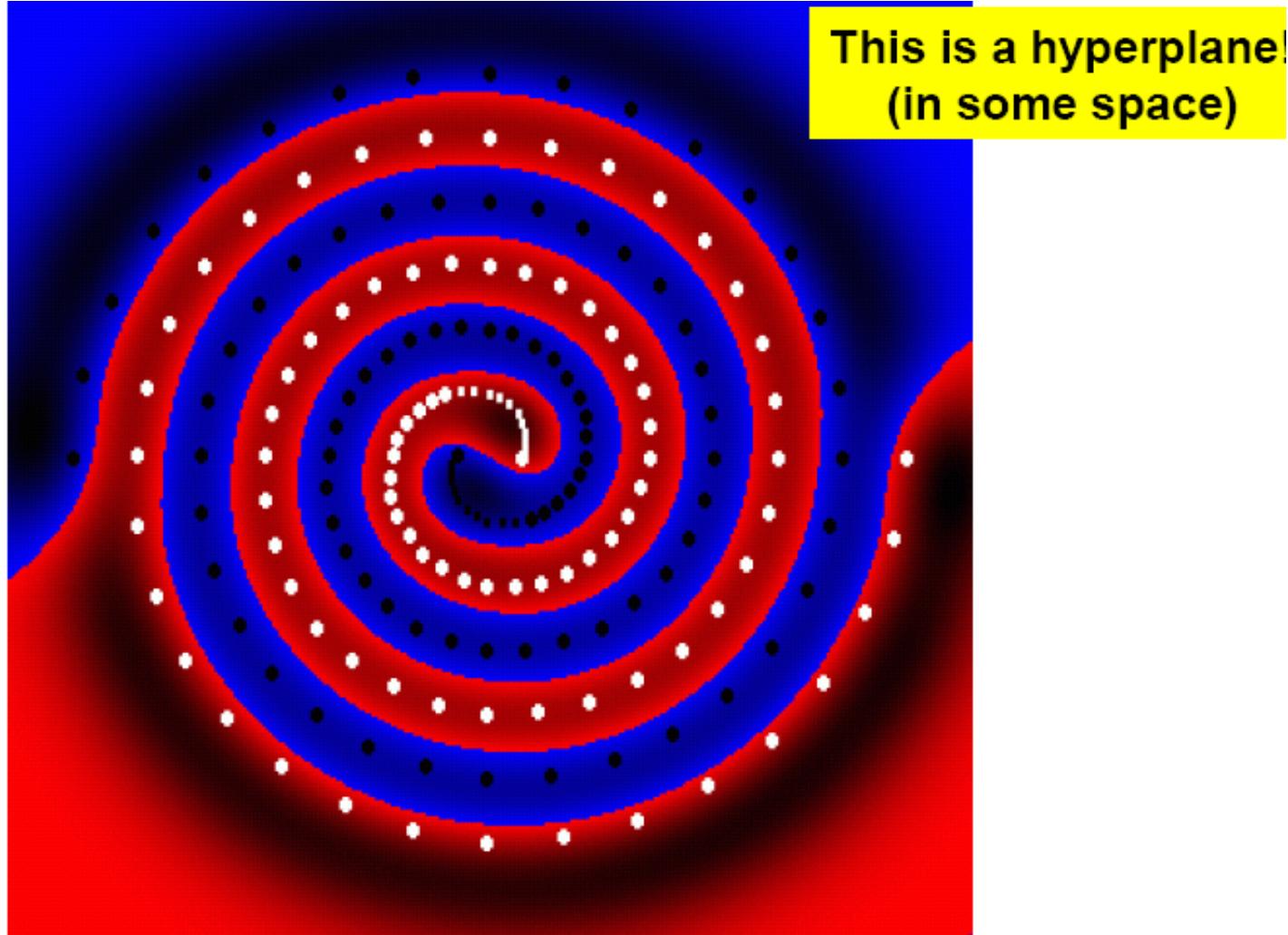
kernels + maximization of the margin with the dual

- pros: kernel
 - problem projected to higher dimension space...
 - not necessarily...
 - but potentially infinite...
 - implicit
 - no need to compute $\phi(x)$...
 - ... which may be unknown!

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle$$

- pros: margin maximization
 - dual doesn't depend on the number of variables
 - minimize the effect of the curse of dimensionality
 - problem remains convex
 - unique solution

gaussian kernel



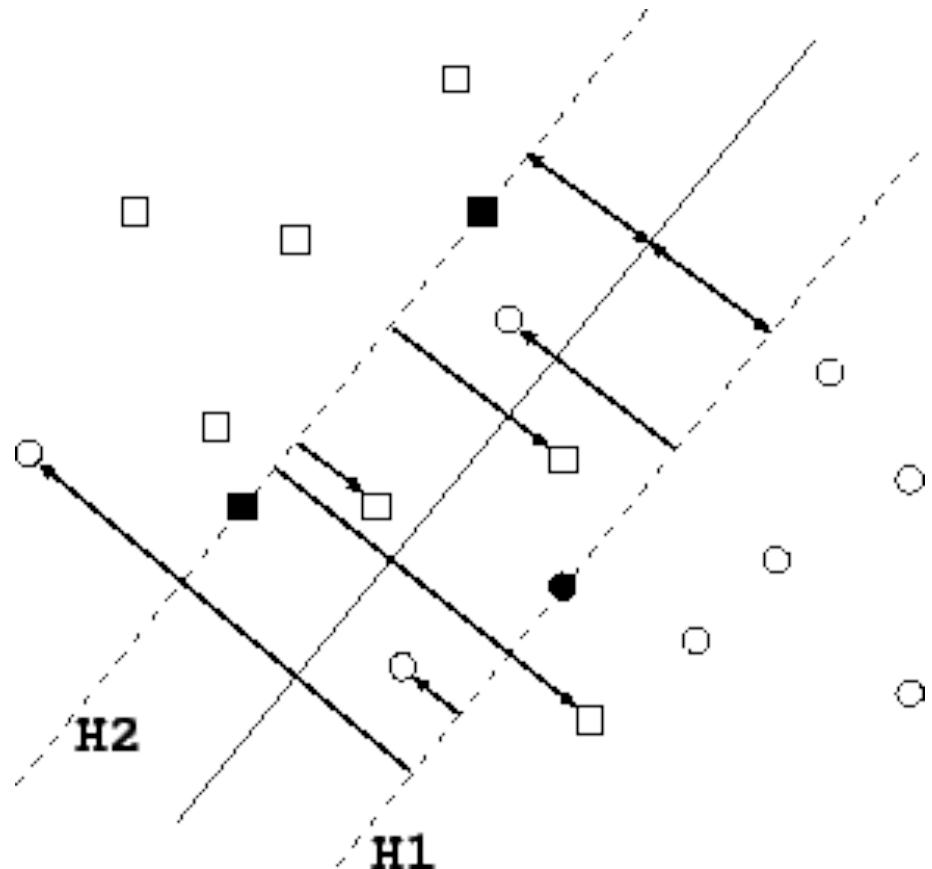
$$e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

relax the objective function

- soft margin
 - maximize the margin
 - minimize error

$$\frac{1}{2} \|\mathbf{w}\|_2 + C \sum_{i=1}^l z_i$$

- C , regularization constant
 - compromise between the importance of the margin and the error
 - yet another parameter...
- most common type of SVM



SVM Soft Margin: dual problem

maximize $\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$

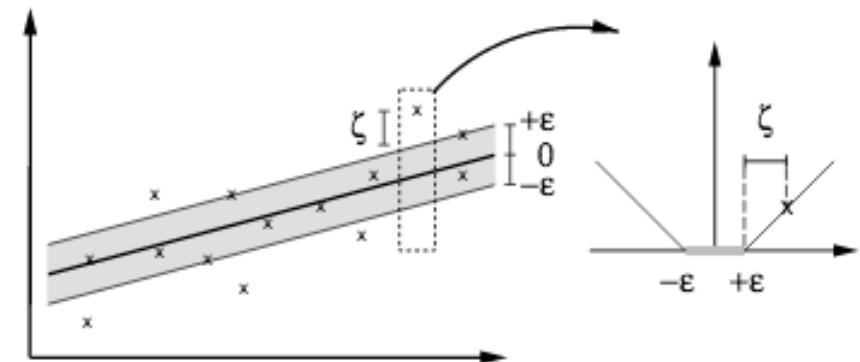
subject to
the constraints

$$\sum_{i=1}^l y_i \alpha_i = 0$$
$$C \geq \alpha_i \geq 0, i = 1, \dots, l$$

- only one difference
 - weight of each example is limited

SVM for Regression

- margin
 - minimize the tube “around” the data
 - Instead of maximizing the distance to closest examples from each class



source: <http://alex.smola.org/papers/2003/SmoSch03b.pdf>

Regression

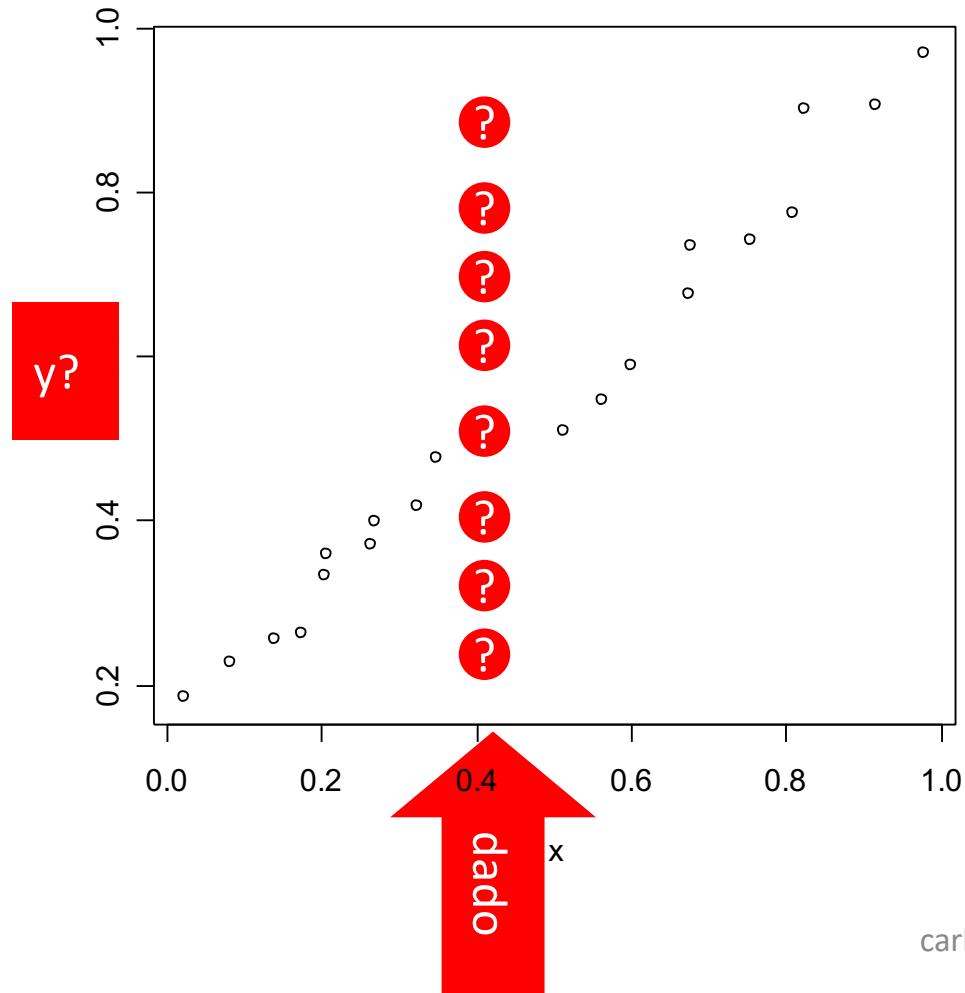
Carlos Soares

(partly using materials from Moreira,
Carvalho & Horvath)

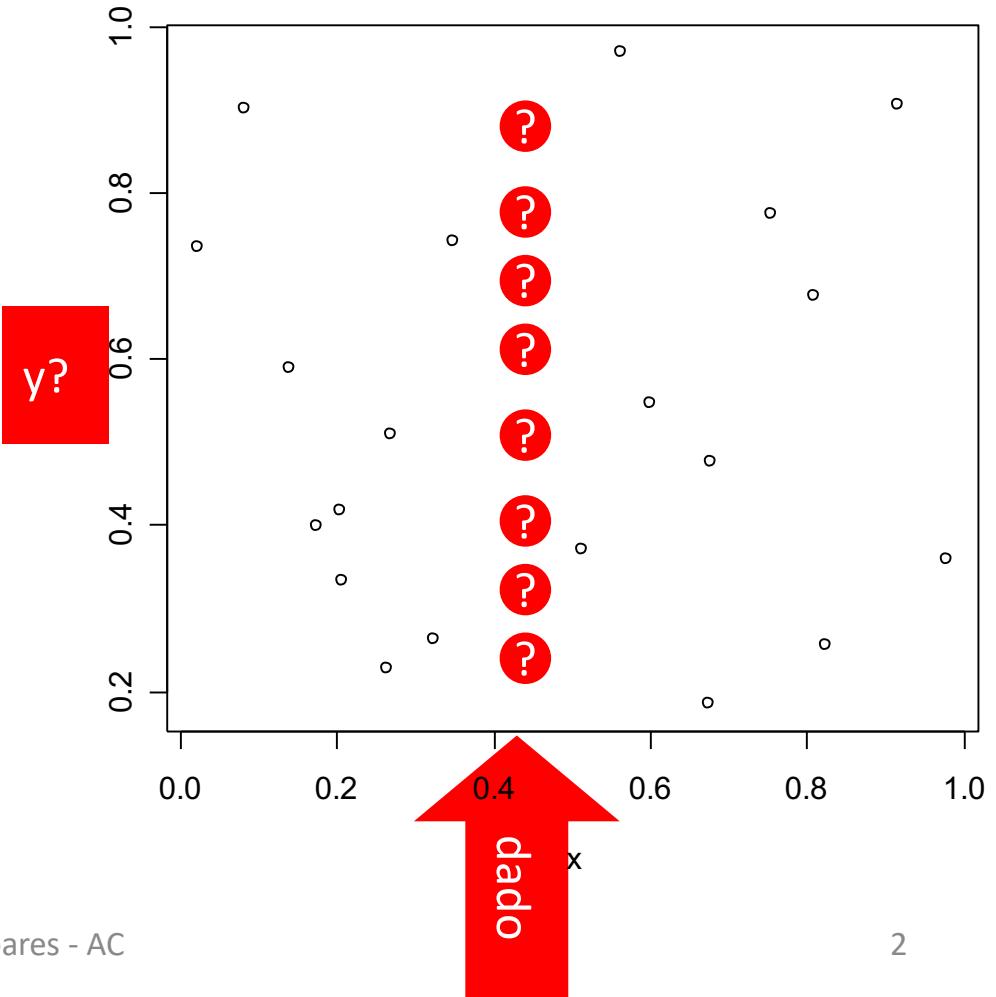


regression

$x = \text{family income}$
 $y = \text{total purchases}$



carlos soares - AC



2

plan & goals

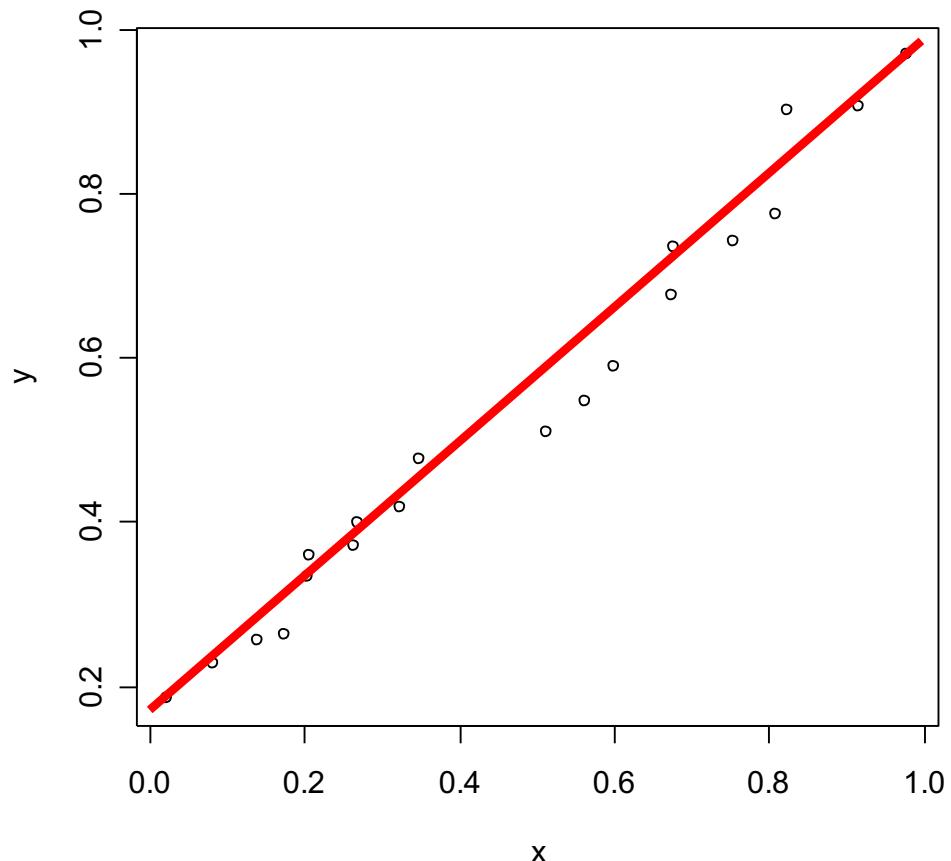
- linear regression
 - interpretation
 - algorithm
- evaluation of regression models
- other algorithms
- regression concepts
 - interpretation of the linear model
 - evaluation measures
- common approaches to adapting learning algorithms for regression

linear regression

- simple case: 2 variables
 x and y

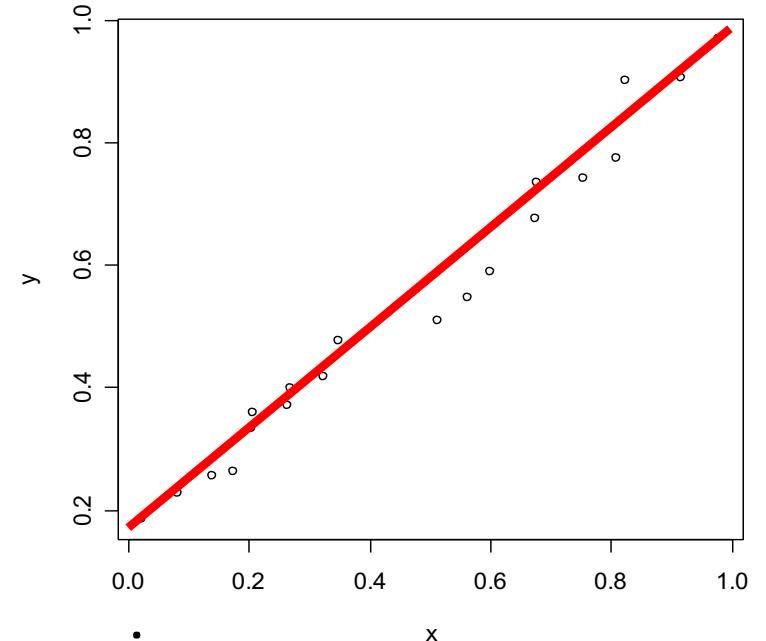
- liner equation

$$\begin{aligned}y &= f(x) \\&= b_0 + b_1 x\end{aligned}$$



interpretation of coefficients

$$y = b_0 + b_1 x$$



- b_0 : intersection of the line with the y axis
 - often hard to interpret
- b_1 : slope of the line
 - variation in the value of y given a 1 unit increase of the value of x

exercise II: analyze linear regression model

- assumes that variables are not correlated
 - influence of each variable is explained separately
 - coefficients are not influenced by changing the set of explanatory variables
 - i.e. attributes
- variation depends on the degree of correlation
 - signal may change!
- ... but empirical results show robustness

Table View Text View Annotations

LinearRegression

```
- 0.108 * CRIM
+ 0.045 * ZN
+ 0.018 * INDUS
+ 2.661 * CHAS
- 17.655 * NOX
+ 3.822 * RM
- 1.459 * DIS
+ 0.304 * RAD
- 0.012 * TAX
- 0.978 * PTRATIO
+ 0.009 * B
- 0.521 * LSTAT
+ 36.696
```

Simple linear regression: estimating parameters

$$y = b_0 + b_1 x$$

$$\hat{b}_1 = \frac{S_{XY}}{S_{XX}}$$

where \hat{b}_1 is an estimate of b

$$S_{XY} = \sum_{i=1}^n [(X_i - \bar{X})(Y_i - \bar{Y})]$$

$$S_{XX} = \sum_{i=1}^n (X_i - \bar{X})^2$$

- \hat{b}_1 should be statistically significantly different from zero
 - if not, there is no meaningful dependency between Y and X
 - this should be tested

$$\hat{b}_0 = \bar{Y} - \hat{\beta} \cdot \bar{X}$$

where \hat{b}_0 is an estimate of b_0

- \hat{b}_0 may or may not be statistically significantly different from zero
 - If not there is no evidence that $Y \neq 0$ when $X=0$.
 - ... which could make sense
 - e.g. value of a customer with 0 income
 - ... or not...
 - e.g. minimum sales of a product without shelf space

Simple linear regression: assumptions

- Linear relationship between x and y
 - also additive
- Errors
 - i.e. unexplained variation in y
 - ... are independently and identically distributed
 - ... homoscedasticity
 - constant variance
 - ... normally distributed

- linear regression
- evaluation of regression models
 - measures
 - methodology
 - bias-variance trade-off
- other algorithms

prediction and evaluation

- given the value of x
- ... the model estimates the value of y

$$\hat{y} = b_0 + b_1 x$$

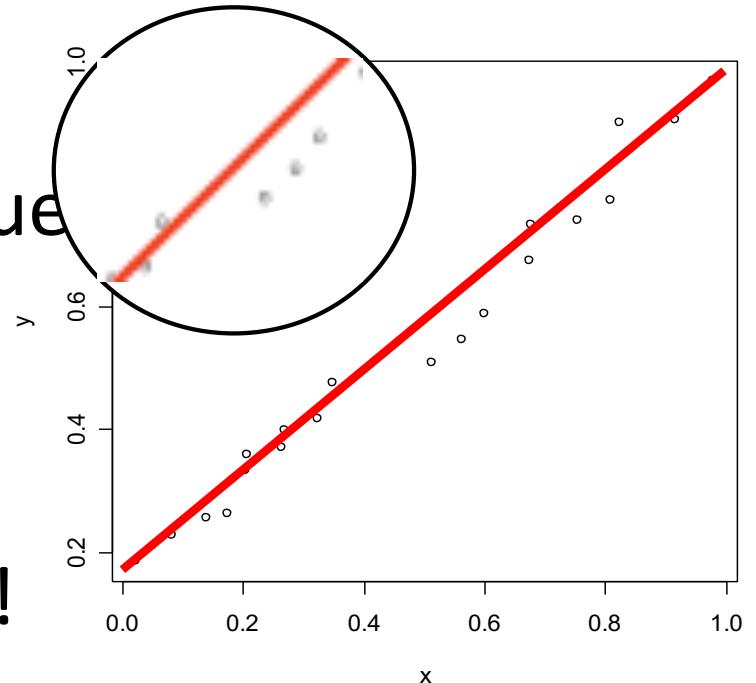
- but the estimate is not perfect!

- erro:

- y : true value

$$\hat{y} - y$$

- \hat{y} : value estimated by the model



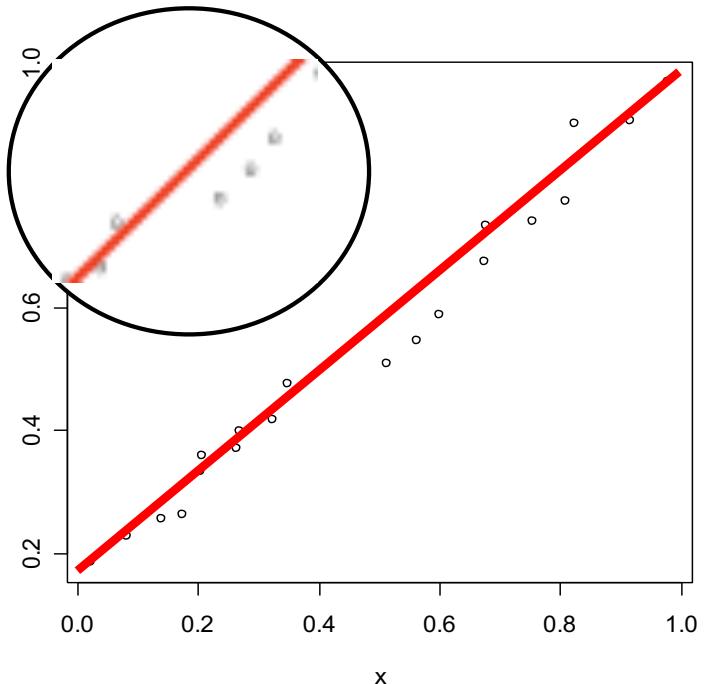
analysis of evaluation measures

- mean error
 - DO NOT USE!
- mean absolute error
 - estimates “typical” error
- mean squared error
 - assigns more weight to larger errorsⁱ
 - ... may be dominated by a few cases
- values depend on the scale of the target variable
 - is the error good or bad?
 - business perspective?
 - does the relationship between x/y represented really exist?

$$\frac{1}{m} \sum_i \hat{y}_i - y_i$$

$$\frac{1}{m} \sum_i |\hat{y}_i - y_i|$$

$$\frac{1}{m} \sum_i (\hat{y}_i - y_i)^2$$



baseline: trivial model

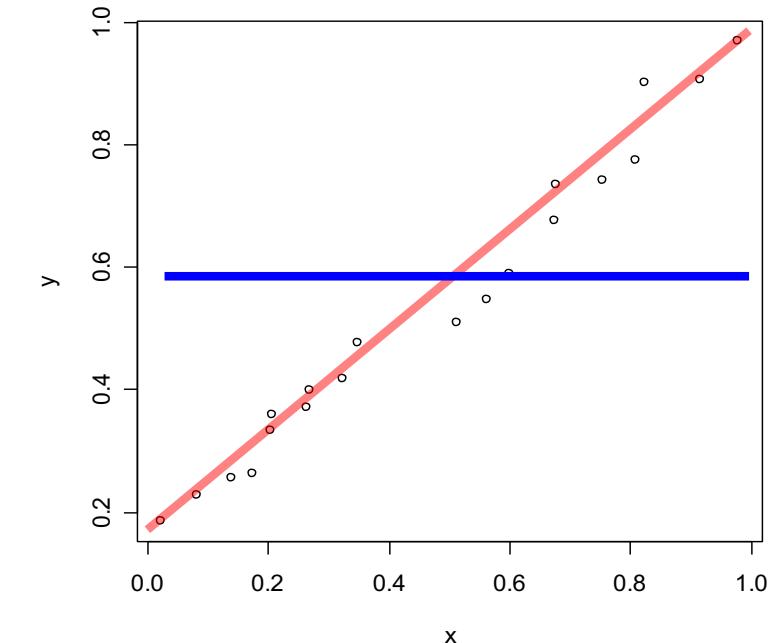
- if we know nothing about the cases
- what is the best prediction we can make?
 - random vs mean

- trivial model

$$\hat{y}_i = \bar{y}$$

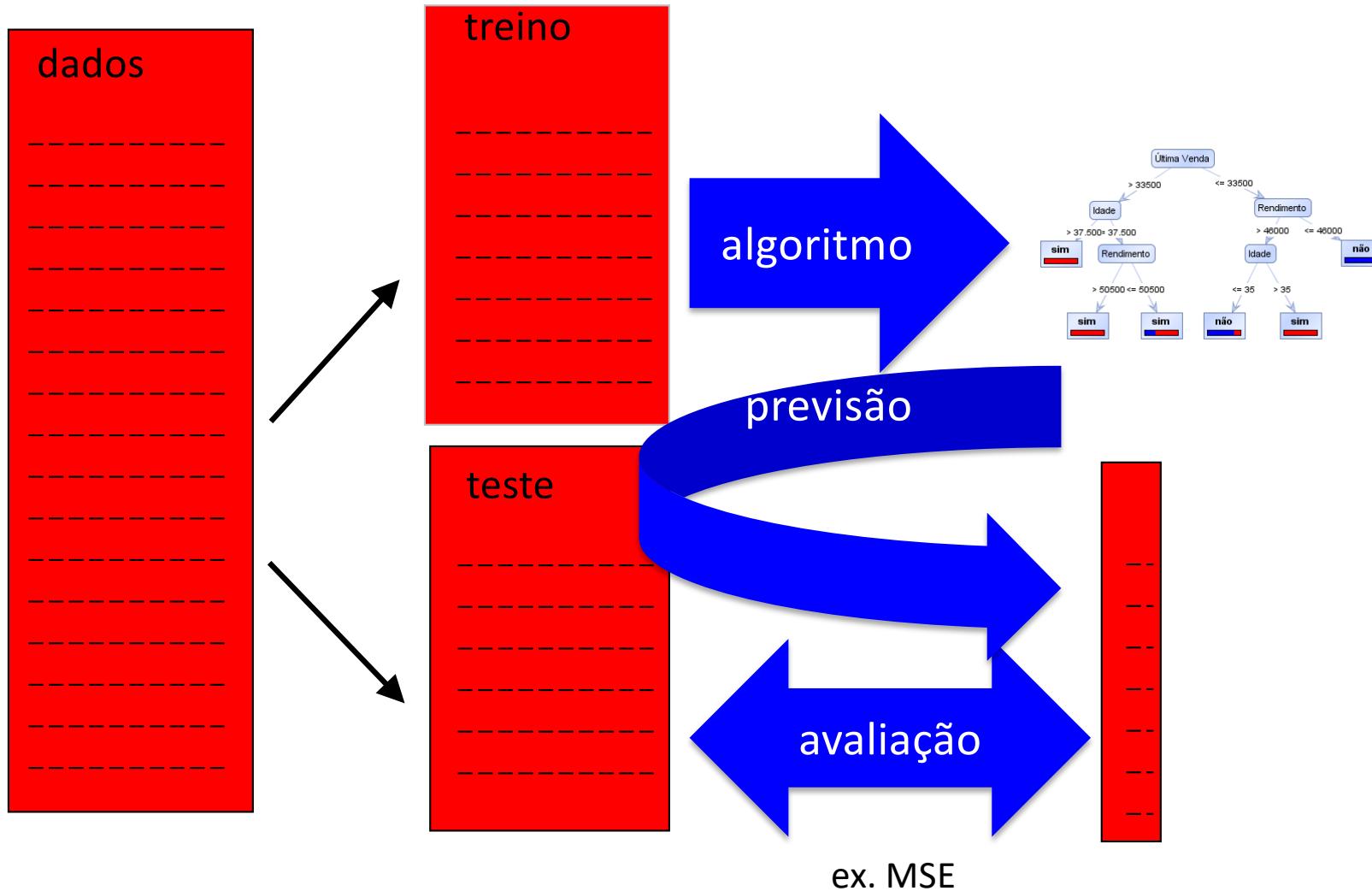
- regression is only useful if its error is lower than the one obtained with the trivial prediction
 - eg. mean squared error

$$\frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$$



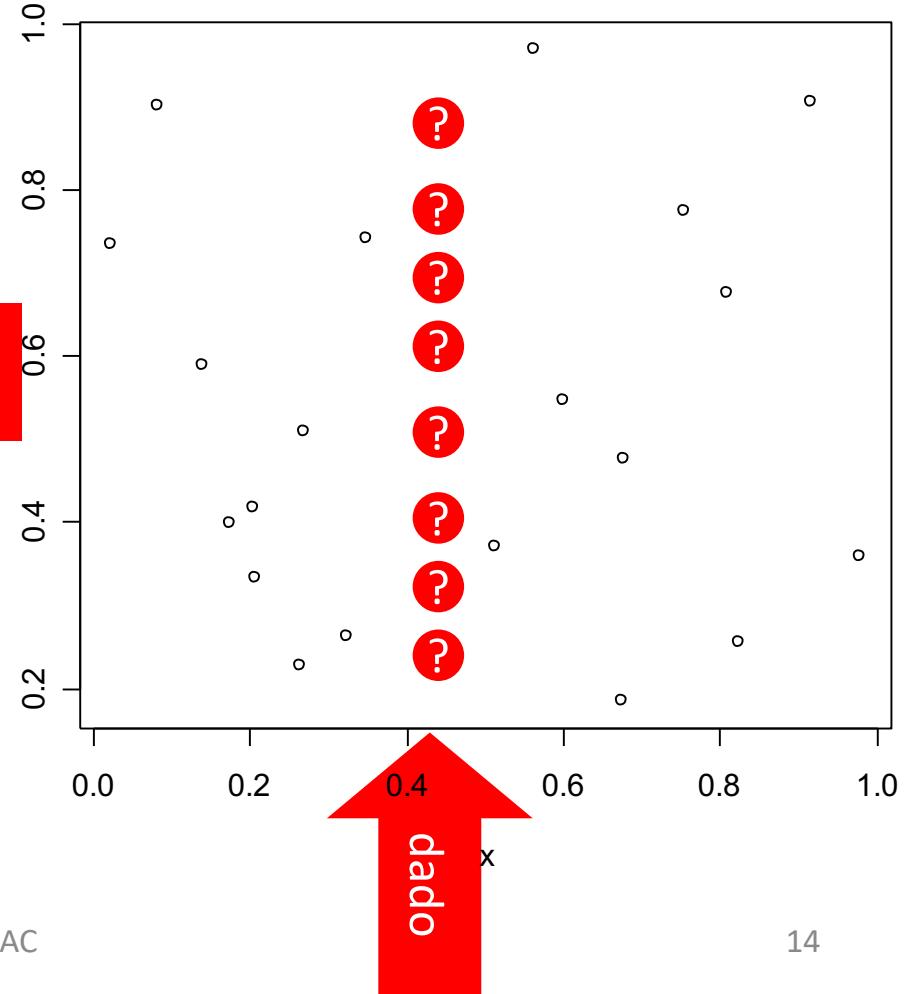
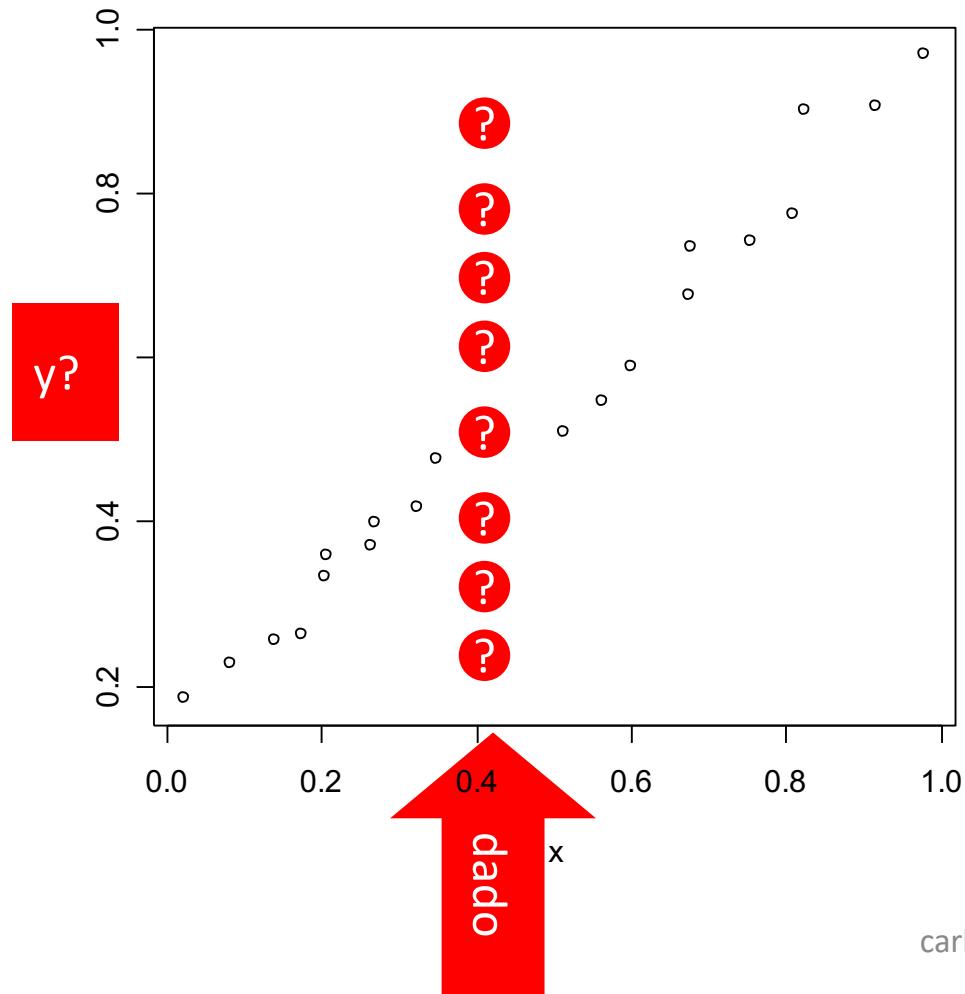
{
0 if regression model is perfect
]0,1[if it is useful
1 if it is equivalent to the trivial model
>1 if it is worse than the trivial model

evaluation methodology: do not forget!



remember?

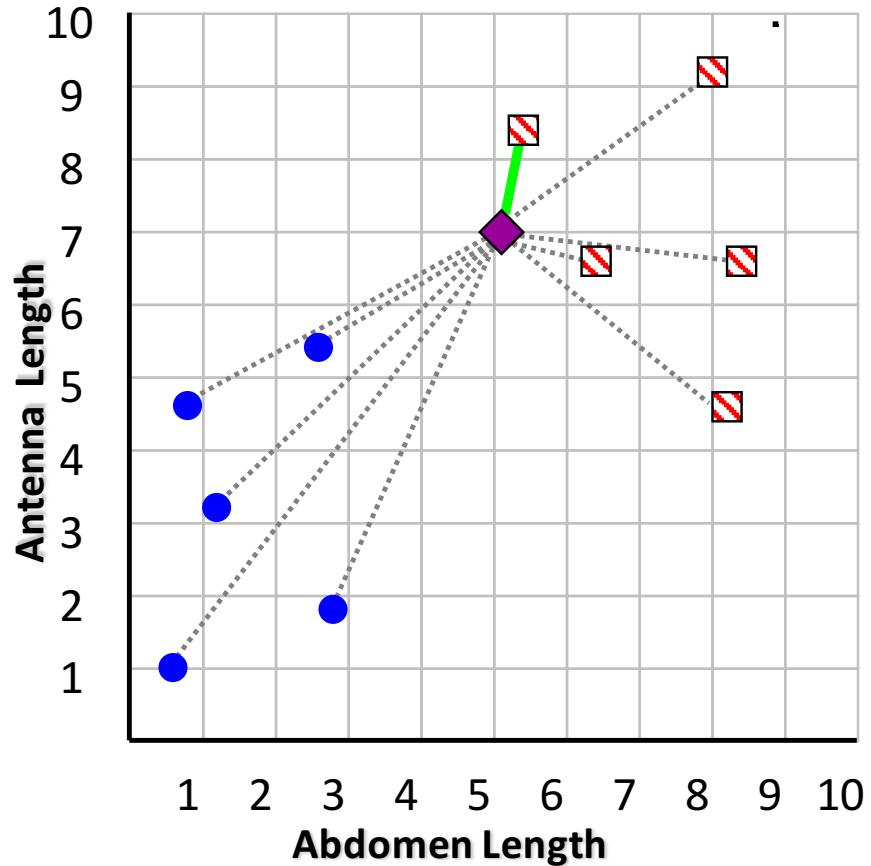
$x = \text{family income}$
 $y = \text{total purchases}$



- linear regression
- evaluation of regression models
- other algorithms
 - kNN
 - trees
 - neural networks
 - support vector machines
 - ... bias & variance

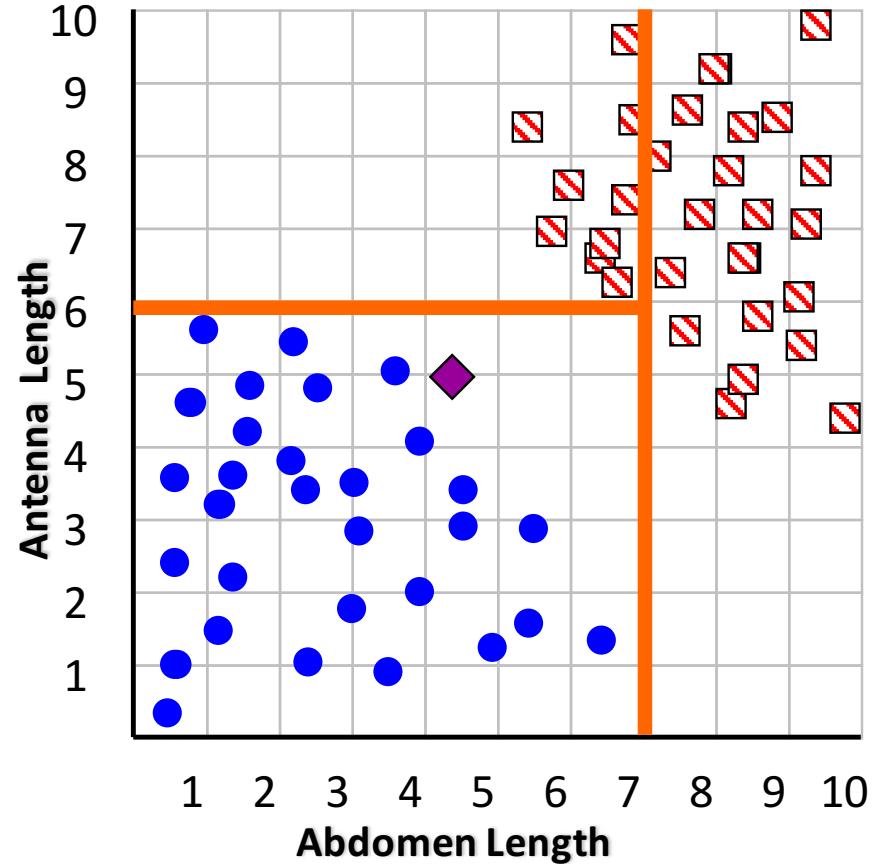
Nearest Neighbor Algorithm for Regression

- find kNN
 - just like for classification
- predict the average of their target values
 - instead of majority voting



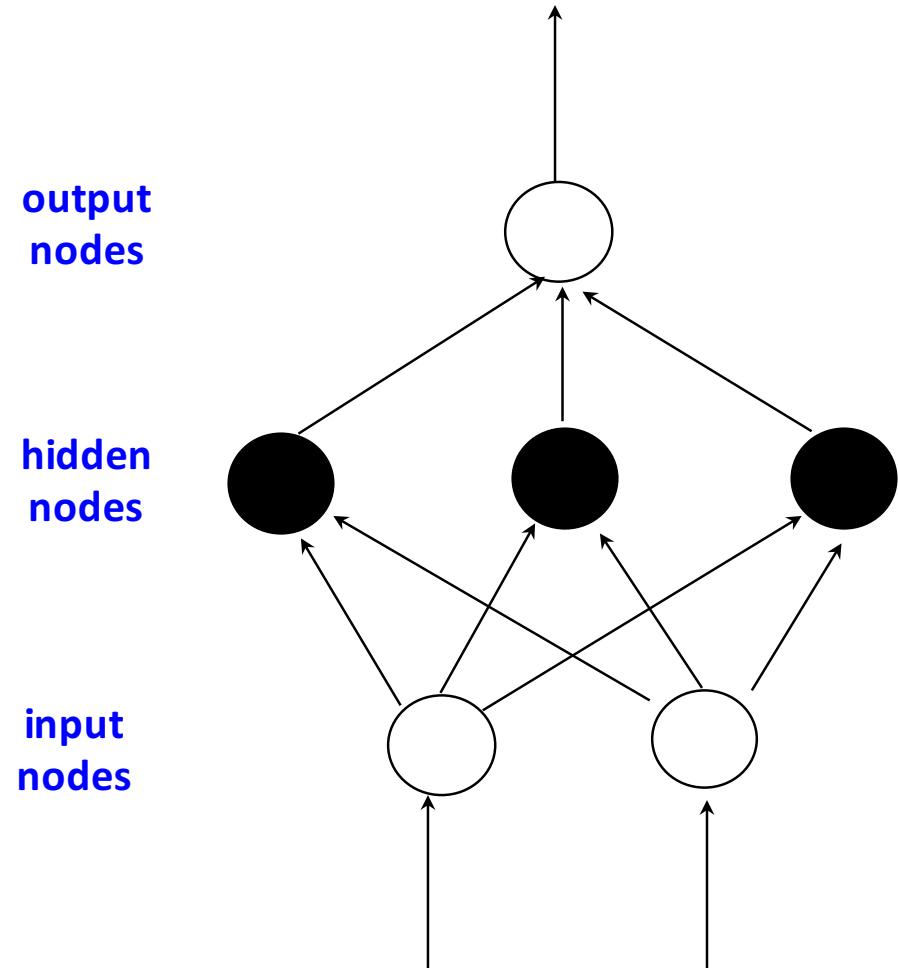
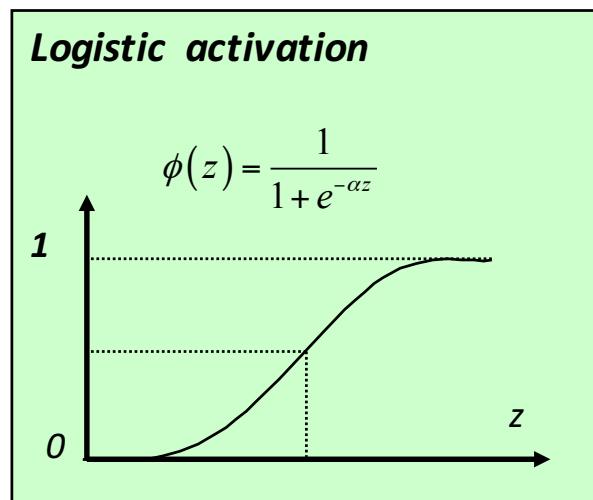
Decision Trees for Regression

- train
 - splitting criterion based on the sum of the variances
 - instead of gini or entropy
- prediction
 - average of targets in the leaf
 - instead of majority voting
- variants
 - model trees
 - using MLR or K-NN in the leaves instead of the average
 - MARS
 - multivariate adaptive regression splines



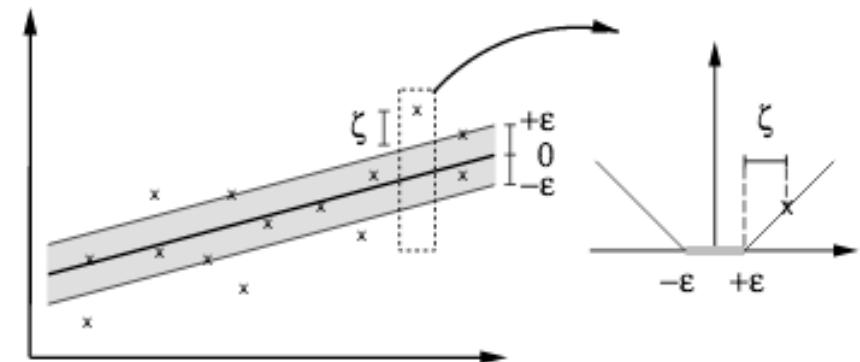
Neural Nets for Regression

- single output node
 - predicted $y = \text{score}$
- continuous activation function
 - e.g. sigmoid
 - also used for classification



SVM for Regression

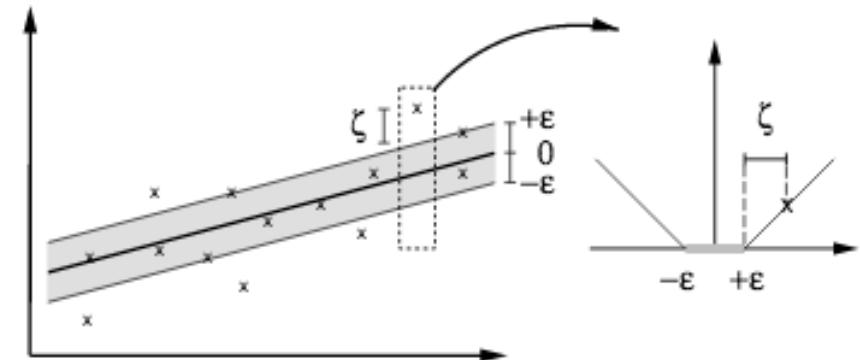
- margin
 - minimize the tube “around” the data
 - Instead of maximizing the distance to closest examples from each class



source: <http://alex.smola.org/papers/2003/SmoSch03b.pdf>

SVM for Regression

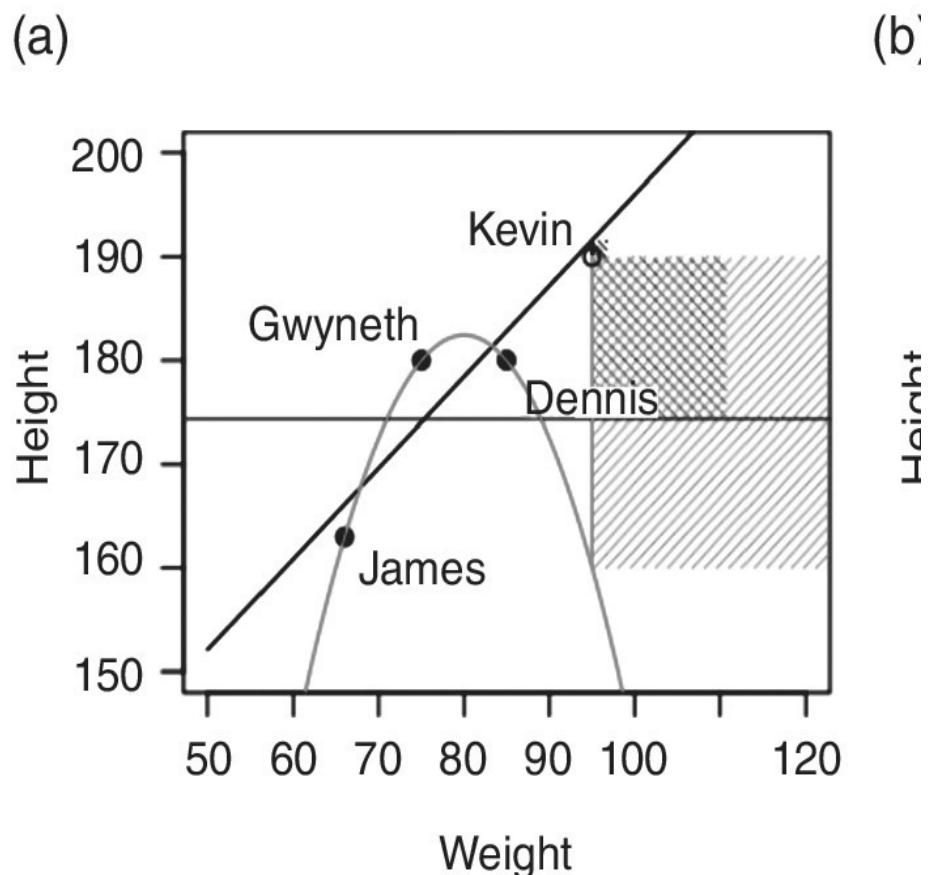
- margin
 - minimize the tube “around” the data
 - Instead of maximizing the distance to closest examples from each class



source: <http://alex.smola.org/papers/2003/SmoSch03b.pdf>

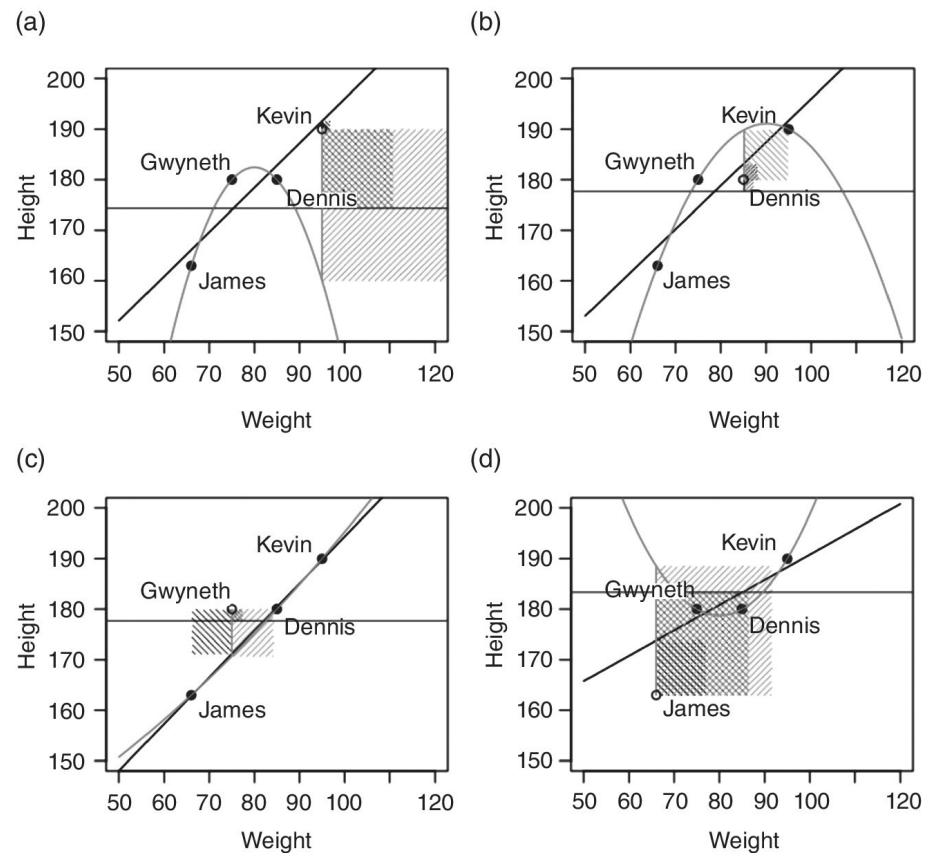
bias

- type of model an algorithm is able to learn given a set of training data
- related to hypothesis language
 - e.g. linear vs quadratic



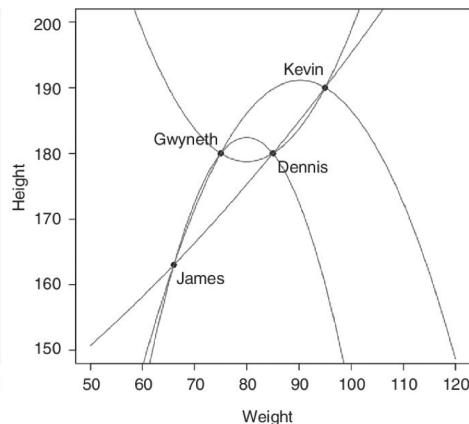
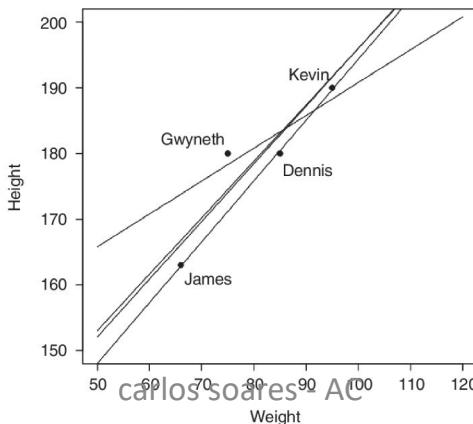
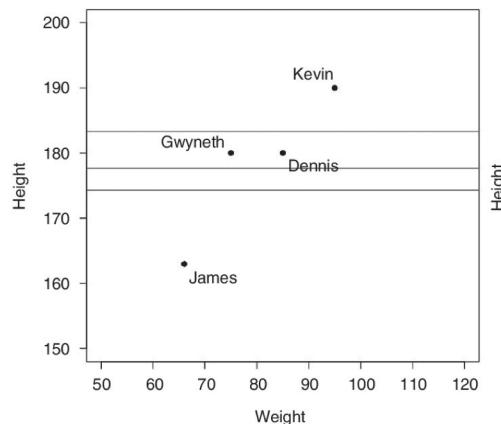
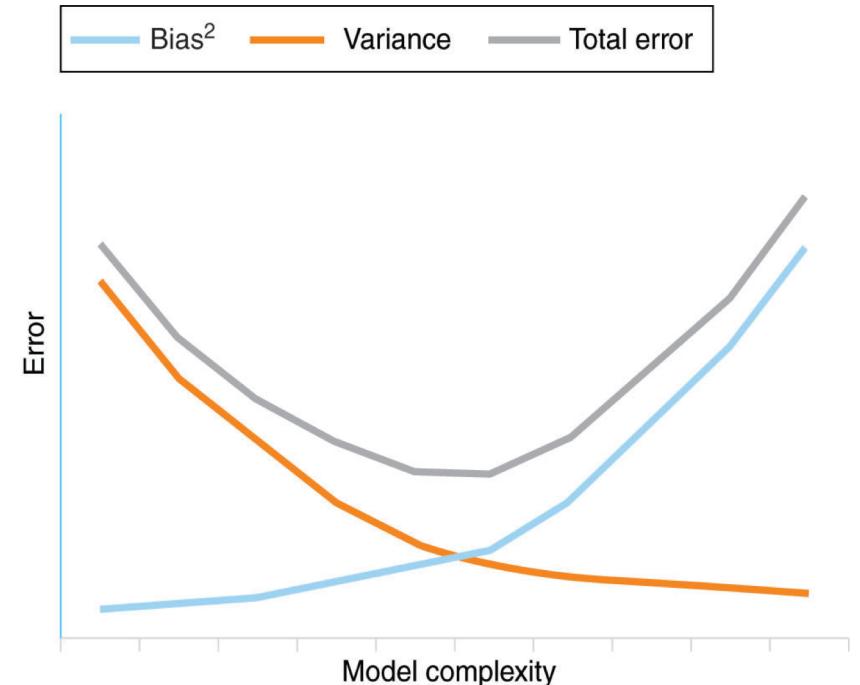
... and variance

- variation in model an algorithm is able to learn, given different training data
 - ie. small changes



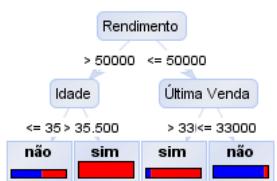
bias-variance trade-off

- Low bias implies high variance and vice-versa
- We would like to find a model with a good trade-off
 - Not too complex but with good predictive power

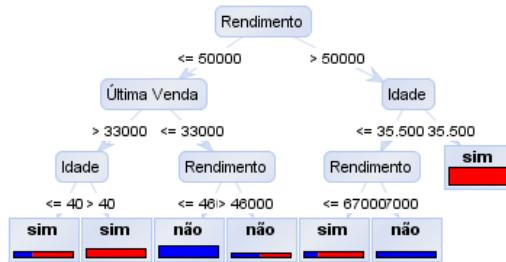


remember overfitting?

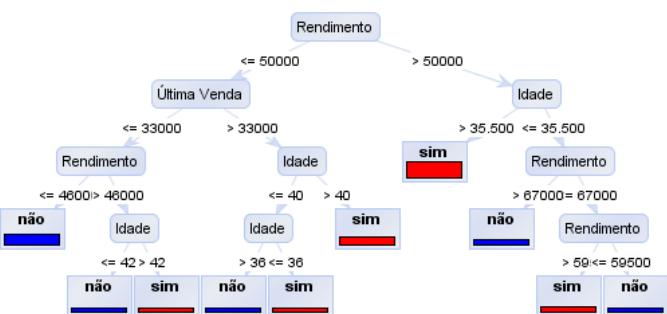
- trees obtained with different values of “minimum leaf size”
 - 4, 2 and 1



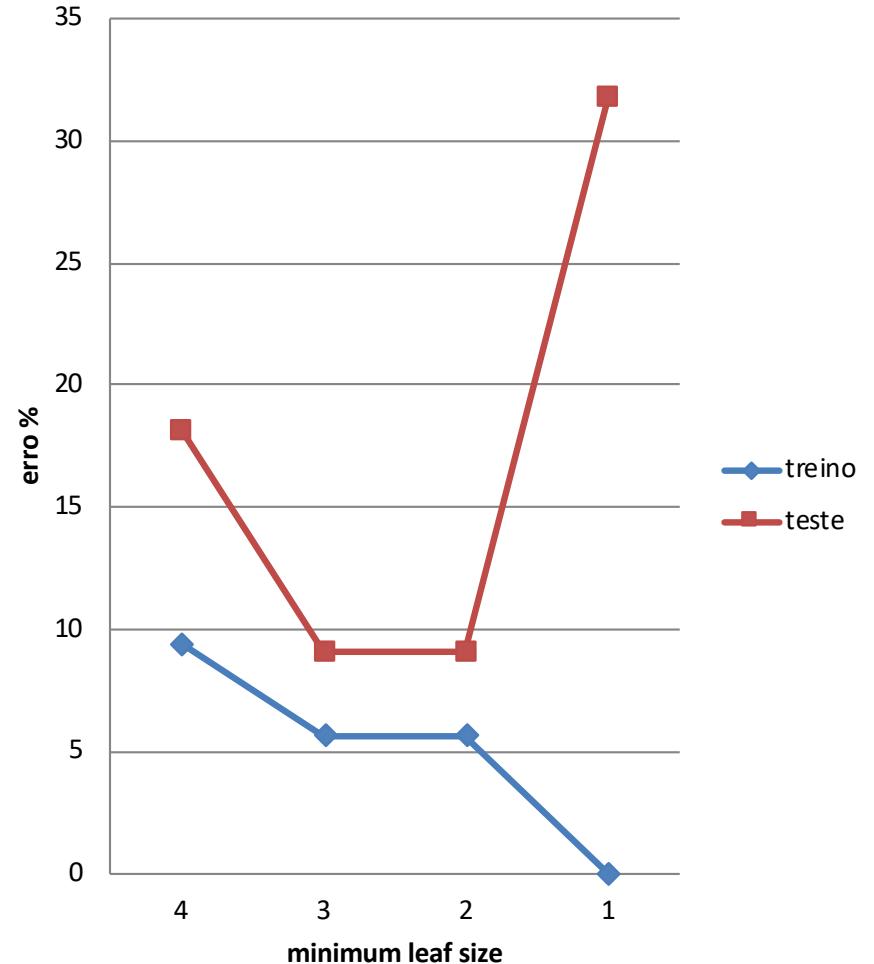
error (train)=18,18%



error (train)=9,09%



error (train)=0,00%

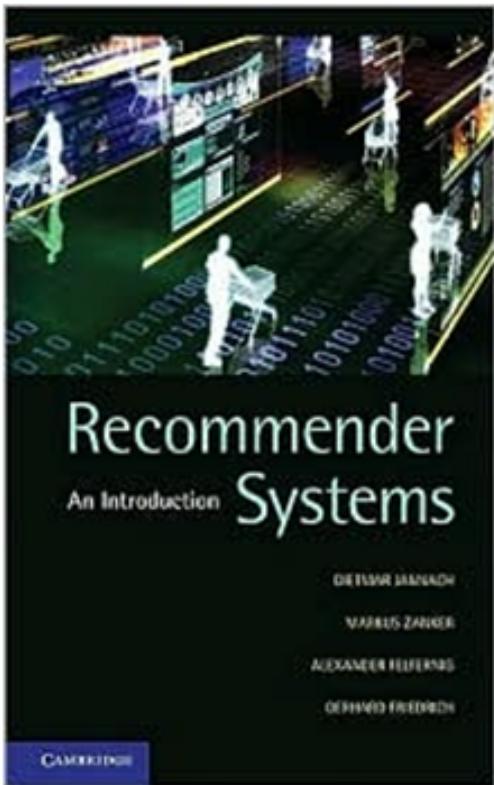


introduction to recommender systems

Carlos Soares
(adapted from materials from the book
“Recommender Systems – An Introduction” by
Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich
Cambridge University Press)

plan

- introduction
 - problem domain
 - purpose and success criteria
 - paradigms of recommender systems
- collaborative filtering
- evaluation
- data
- more algorithms
- challenges



Recommender Systems: An Introduction

by [Dietmar Jannach](#), [Markus Zanker](#), [Alexander Felfernig](#), [Gerhard Friedrich](#)

AVERAGE CUSTOMER RATING:

([Be the first to review](#))

Gefällt mir



Registrieren, um sehen zu können, was
deinen Freunden gefällt.

FORMAT:

Hardcover

NOOKbook (eBook) - not available

[Tell the publisher you want this in NOOKbook format](#)

NEW FROM BN.COM

\$65.00 List Price

\$52.00 Online Price
(You Save 20%)

Add to Cart

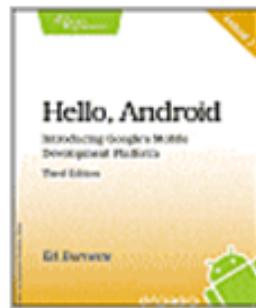
NEW & USED FROM OUR

New starting at **\$56.46** (You Save 20%)
Used starting at **\$51.98** (You Save 20%)

See All Prices

Table of Contents

Customers who bought this also bought



RS @ ACD = CS



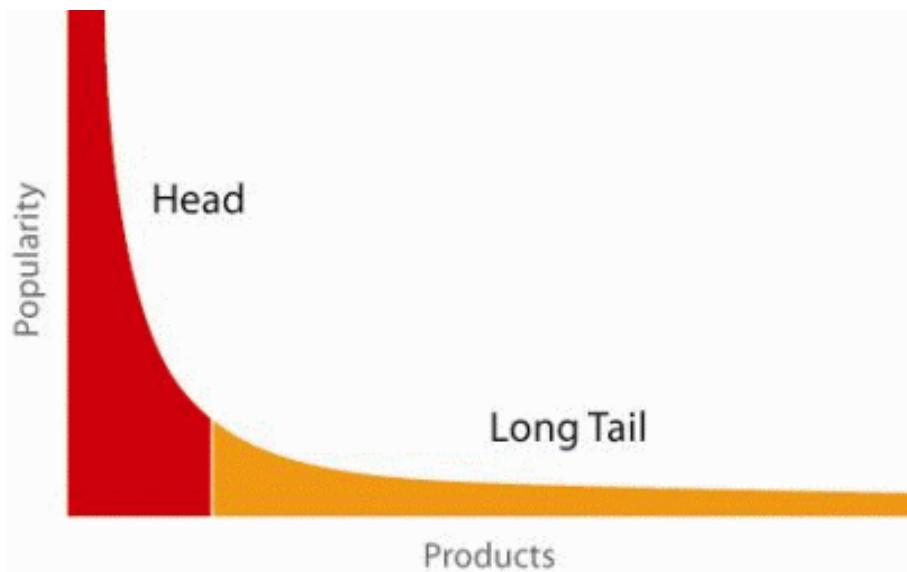
Purpose and success criteria (1/2)

- Retrieval
 - Users know in advance what they want
 - Provide "correct" proposals
 - Reduce search costs
- Recommendation
 - Items unknown to users
 - Serendipity
- Prediction
 - Predict to what degree users like an item
- Interaction
 - Give users a "good feeling"
 - Convince/persuade users - explain
- Conversion
 - Increase "hit", "clickthrough", "lookers to bookers" rates
 - Optimize sales margins and profit

Serendipity and the Long Tail

[or why the best place to hide a dead body is
the 2nd page of results of google search]

- Recommend widely unknown items that users might actually like!
- 20% of items accumulate 74% of all positive ratings



Recommender systems: definition

- Given
 - User model
 - e.g. ratings, preferences, demographics, situational context
 - Items
 - with or without description of item characteristics
- Find
 - Relevance score
 - Typically used for ranking
- Relation to Information Retrieval
 - IR is finding material [...] of an unstructured nature [...] that satisfies an information need from within large collections [...].

(1) Manning, Raghavan, and Schütze, *Introduction to information retrieval*, Cambridge University Press, 2008

Paradigms of recommender systems

Recommender systems reduce information overload by estimating relevance



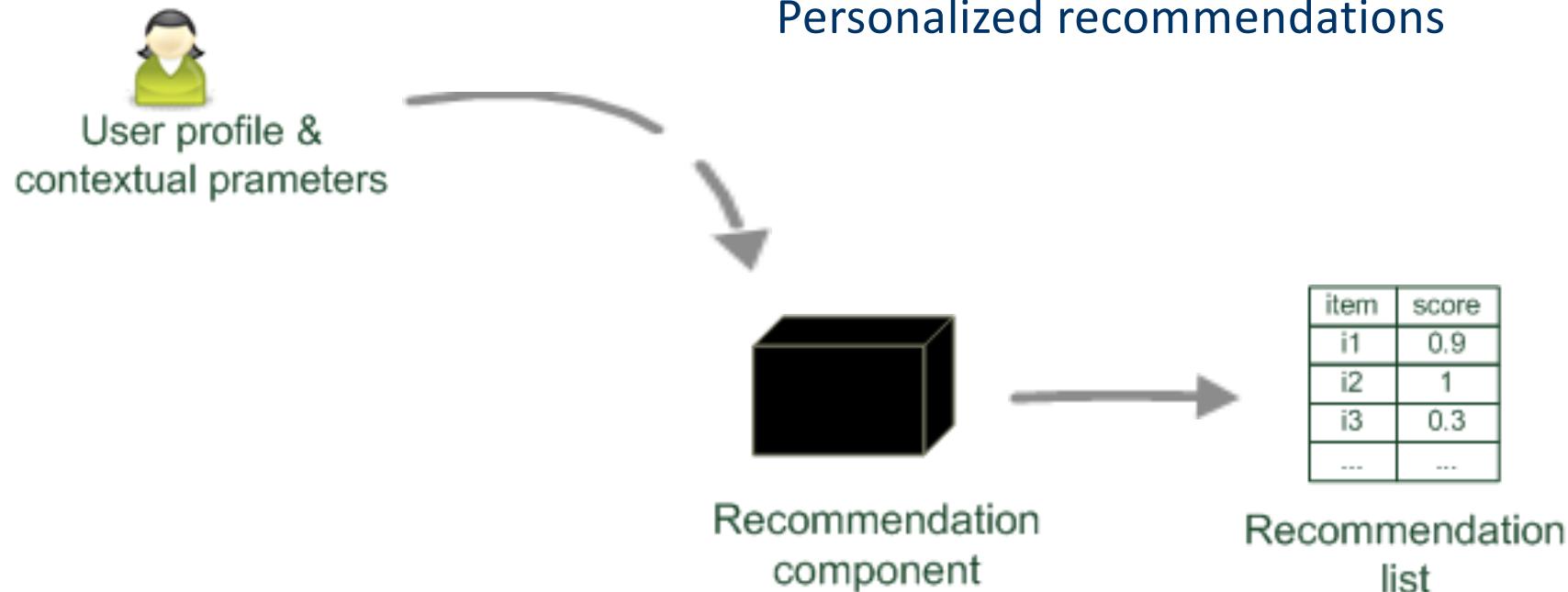
Recommendation component



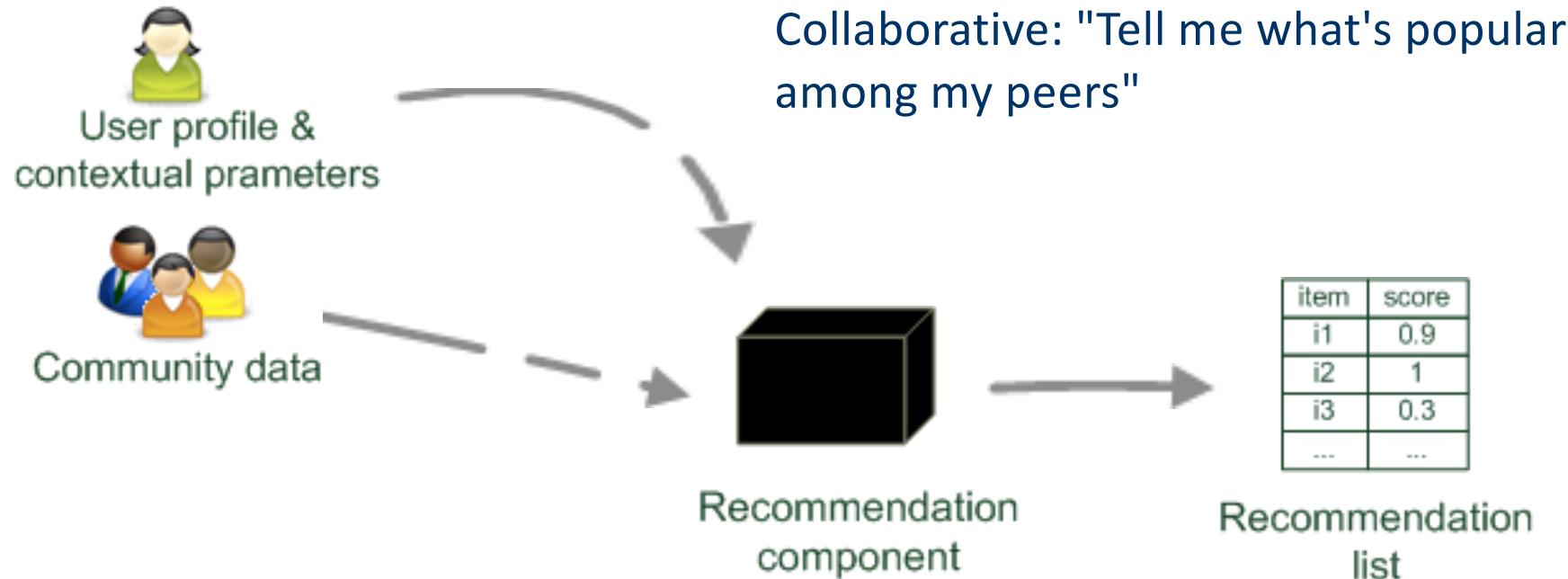
item	score
i1	0.9
i2	1
i3	0.3
...	...

Recommendation list

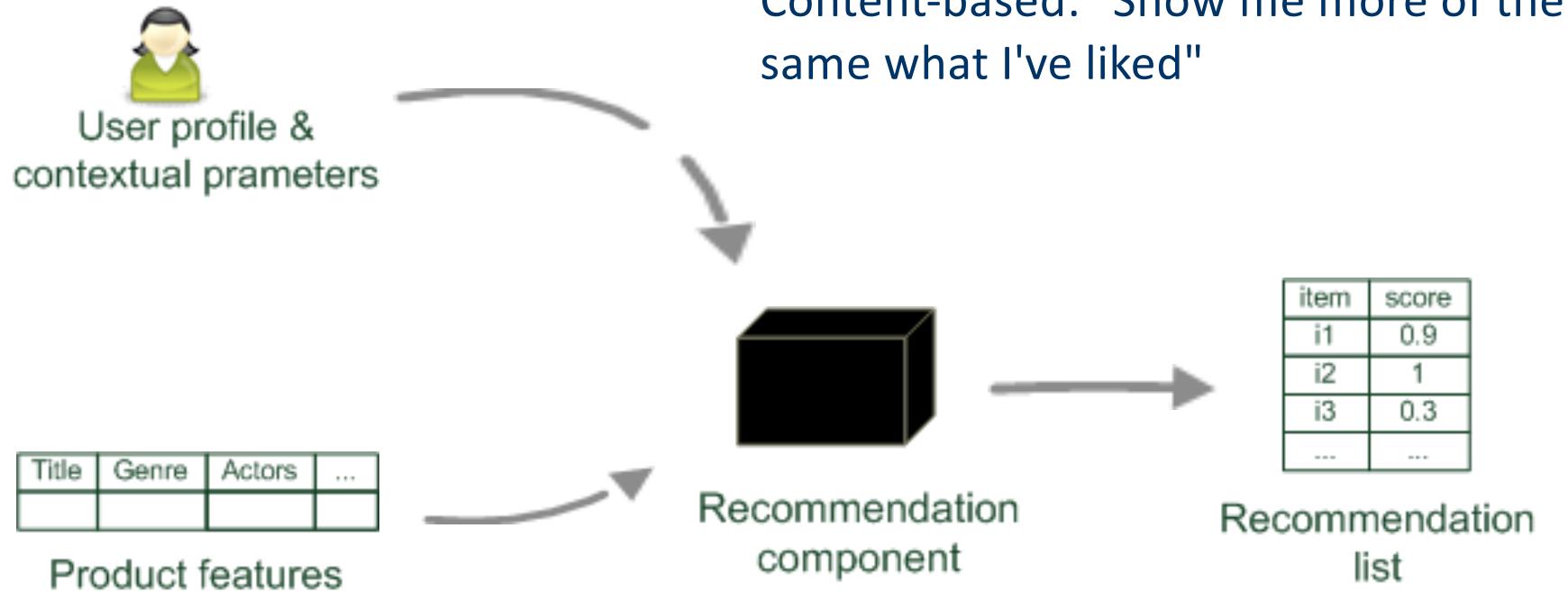
Paradigms of recommender systems



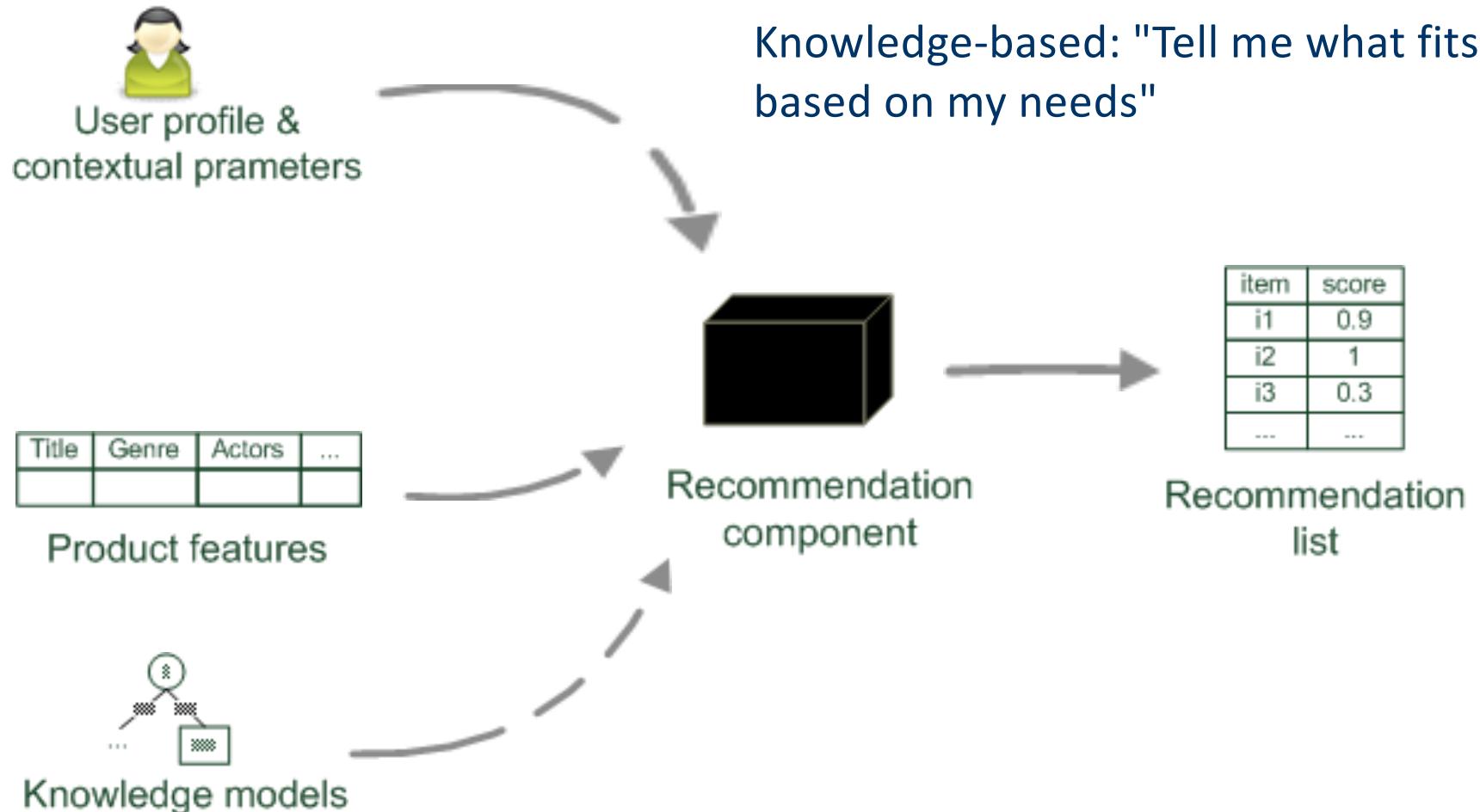
Paradigms of recommender systems



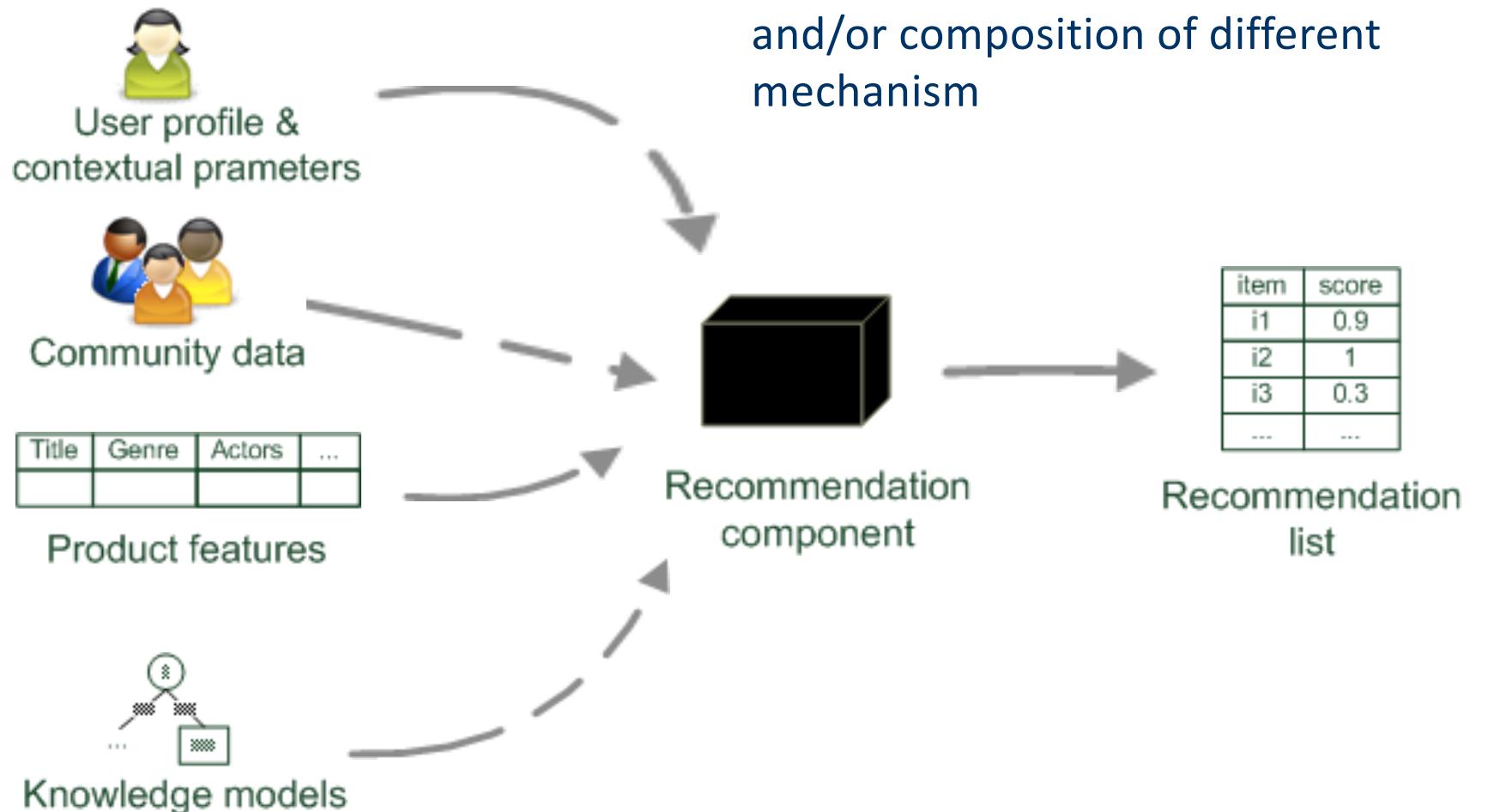
Paradigms of recommender systems



Paradigms of recommender systems



Paradigms of recommender systems



- introduction
- collaborative filtering
 - pure CF approaches
 - user-based nearest-neighbor
- evaluation
- data
- more algorithms
- challenges



pure CF approaches

- Input
 - matrix of given user-item ratings
- Output types
 - degree to what the current user will like or dislike a certain item
 - a top-N list of recommended items

User\Item	Item 1	Item 2	Item 3	Item 4	Item N-2	Item N-1	Item N
User 1	1		1						
User 2				1				1	
User 3	1		1		1		1		
User 4									1
...									
...		1			1				
User M-2						1			
User M-1								1	
User M		1					1		

User-based nearest-neighbor CF

- basic algorithm
 - given an "active user" u and an item i not yet seen by u
 - find a set of users (peers/nearest neighbors) who liked the same items as u in the past and who have rated item i
 - combine their ratings to predict, if u will like item i
 - e.g. average
 - ... repeat do this for all items that u has not seen
 - recommend the best-rated items
- basic assumptions
 - if users had similar tastes in the past they will have similar tastes in the future
 - user preferences remain stable and consistent over time

User-based nearest-neighbor CF: example

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

$$pred(u, p) = \bar{r}_a + \frac{\sum_{b \in N} sim(u, b) * (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(u, b)}$$

- calculate whether the neighbors' ratings for the unseen item i are higher or lower than their average rating
- ... weight, using the similarity with the active user, u , as a weight
- add/subtract the active user's average rating

- introduction
- collaborative filtering
- evaluation
 - introduction
 - offline evaluation
 - metrics
 - online evaluation
- data
- more algorithms
- challenges

evaluate, we must...

- business questions
 - do customers like/buy recommended items?
 - do customers buy items they otherwise would have not?
 - are they satisfied with a recommendation after purchase?
- ... lead to empirical evaluation
 - is the approach efficient with respect to a specific criteria like accuracy, user satisfaction, response time, serendipity, online conversion,
- ... during development and in deployment

... because the no-free-lunch theorem is out to get you!

- many techniques available
 - [will be discussed in the next lesson]
- SO...
 - which one is the best in a given application domain?
 - what are the success factors of different techniques?
 - comparative analysis based on an optimality criterion?

offline evaluation method

- data
 - collected in your problem
 - benchmark datasets

User\Item	Item 1	Item 2	Item 3	Item 4	Item N-2	Item N-1	Item N
User 1	1		1						
User 2				1				1	
User 3	1		1		1		1		
User 4									1
...									
...		1		1					
User M-2						1			
User M-1								1	
User M		1				1			

train and test

- **training set**
 - randomly selected share of known **ratings**
 - build the model
- **testing set**
 - remaining share of withheld ratings
 - ground truth to evaluate the model's quality
 - ... by comparing with its recommendations
- perhaps taking time into account rather than randomly

User\Item	Item 1	Item 2	Item 3	Item 4	Item N-2	Item N-1	Item N
User 1	1		1						
User 2				1				1	
User 3	1		1					1	
User 4				1					1
...									
...	1				1				
User M-2						1			
User M-1							1		
User M		1					1		

... maybe with a twist

- **training set**
 - randomly selected share of known **users**
 - build the model
- **testing set**
 - remaining share of withheld users
 - recommendations based on **observed items**
 - ... compared to **hidden items**
- perhaps taking time into account rather than randomly
- ... and removing **useless data**

User\Item	Item 1	Item 2	Item 3	Item 4	Item N-2	Item N-1	Item N
User 1	1		1						
User 2				1				1	
User 3	1		1		1		1		
User 4									1
...	1			1					
User M-2						1			
User M-1							1		
User M		1				1			

Metrics

- borrowing from information retrieval (IR)

		Reality	
		Actually Good	Actually Bad
Prediction	Rated Good	True Positive (tp)	False Positive (fp)
	Rated Bad	False Negative (fn)	True Negative (tn)

All recommended items

All good items

Precision and Recall

- Recommendation is viewed as information retrieval task
 - i.e. retrieve (recommend) all items which are predicted to be “good”
- **Precision:** a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved
 - E.g. the proportion of recommended movies that are actually good

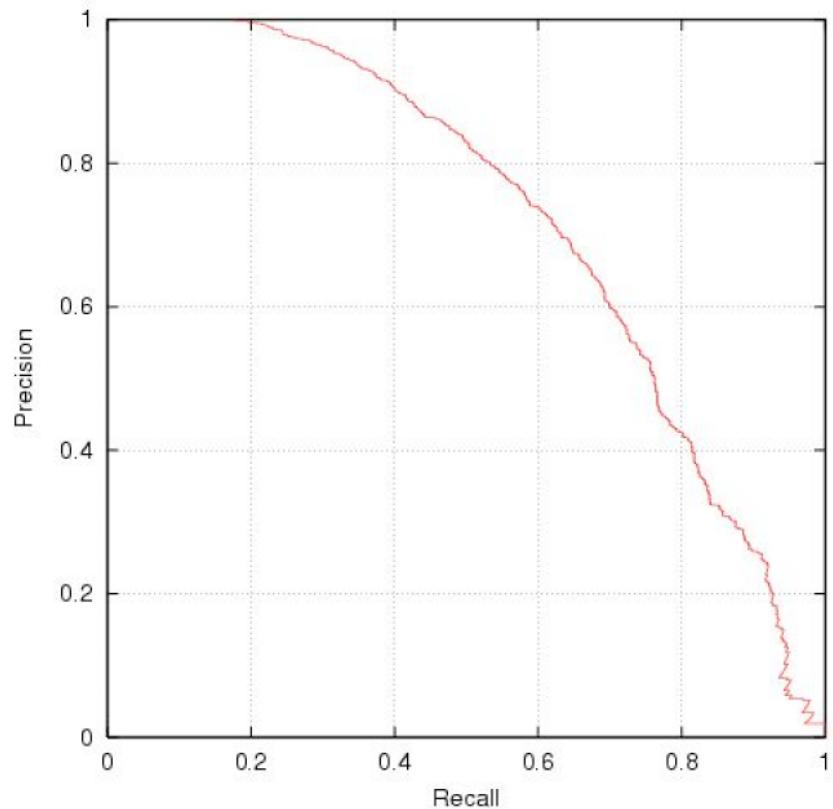
$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|\text{all recommendations}|}$$

- **Recall:** a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items
 - E.g. the proportion of all good movies recommended

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|\text{all good movies}|}$$

ROC, AUC and friends

- typically when a recommender system is tuned to increase precision, recall decreases as a result
 - or vice versa



F_1 Metric

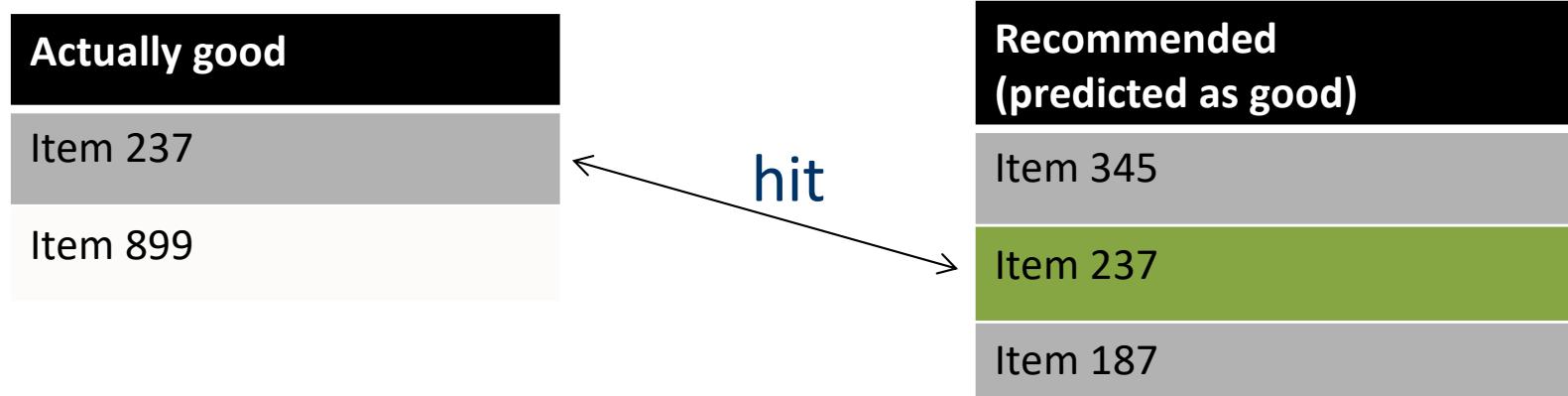
- The **F_1 Metric** attempts to combine Precision and Recall into a single value for comparison purposes.
 - May be used to gain a more balanced view of performance

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- The F_1 Metric gives equal weight to precision and recall
 - Other F_β metrics weight recall with a factor of β .

ranks matter!

For a user:



- **Rank metrics** extend recall and precision to take the positions of correct items in a ranked list into account
 - Relevant items are more useful when they appear earlier in the recommendation list
 - Particularly important in recommender systems as lower ranked items may be overlooked by users

Rank Score

- extends the recall metric to take the positions of correct items in a ranked list into account
- the ratio of the Rank Score of the correct items to best theoretical Rank Score achievable for the user

$$rankscore = \frac{rankscore_p}{rankscore_{\max}}$$

$$rankscore_p = \sum_{i \in h} 2^{-\frac{\text{rank}(i)-1}{\alpha}}$$

$$rankscore_{\max} = \sum_{i=1}^{|T|} 2^{-\frac{i-1}{\alpha}}$$

Where:

- h is the set of correctly recommended items, i.e. hits
- rank returns the position (rank) of an item
- T is the set of all items of interest
- α is the *ranking half life*, i.e. an exponential reduction factor

Metrics: Normalized Discounted Cumulative Gain

- Discounted cumulative gain (DCG)
 - Logarithmic reduction factor

$$DCG_{pos} = rel_1 + \sum_{i=2}^{pos} \frac{rel_i}{\log_2 i}$$

Where:

- *pos* denotes the position up to which relevance is accumulated
- *rel_i* returns the relevance of recommendation at position *i*

- Idealized discounted cumulative gain (IDCG)
 - Assumption that items are ordered by decreasing relevance

$$IDCG_{pos} = rel_1 + \sum_{i=2}^{|h|-1} \frac{rel_i}{\log_2 i}$$

- Normalized discounted cumulative gain (nDCG)
 - Normalized to the interval [0..1]

$$nDCG_{pos} = \frac{DCG_{pos}}{IDCG_{pos}}$$

Example

- Assumptions:

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$- |T| = 3$$

$$- \text{Ranking half life (alpha)} = 2$$

$$\text{rankscore} = \frac{\text{rankscore}_p}{\text{rankscore}_{\max}} \approx 0.71$$

$$\text{rankscore}_p = \frac{1}{2^{\frac{2-1}{2}}} + \frac{1}{2^{\frac{3-1}{2}}} + \frac{1}{2^{\frac{4-1}{2}}} = 1.56$$

$$\text{rankscore}_{\max} = \frac{1}{2^{\frac{1-1}{2}}} + \frac{1}{2^{\frac{2-1}{2}}} + \frac{1}{2^{\frac{3-1}{2}}} = 2.21$$

$$nDCG_5 \frac{DCG_5}{IDCG_5} \approx 0.81$$

$$DCG_5 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} = 2.13$$

$$IDCG_5 = 1 + \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 2.63$$

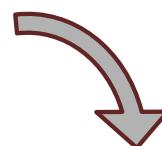
Rankscore (exponential reduction) < NDCG (log. red.)

Average Precision

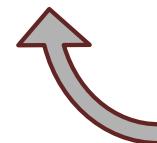
- ranked precision metric that places emphasis on highly ranked correct predictions (hits)
 - average of precision values determined after each successful prediction, i.e.

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$AP = \frac{1}{3} \left(\frac{1}{1} + \frac{2}{4} + \frac{3}{5} \right) = \frac{21}{30} = 0.7$$



$$AP = \frac{1}{3} \left(\frac{1}{2} + \frac{2}{3} + \frac{3}{4} \right) = \frac{23}{36} \approx 0.639$$



Rank	Hit?
1	X
2	
3	
4	X
5	X

metrics for rating prediction

- ground truth = ratings
 - i.e. regression problem
- Mean Absolute Error (MAE) computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - r_i|$$

- Root Mean Square Error (RMSE) is similar to MAE, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - r_i)^2}$$

- introduction
- collaborative filtering
- evaluation
 - introduction
 - offline evaluation
 - metrics
 - online evaluation
- data
- more algorithms
- challenges

online evaluation: example

- Effectiveness of different algorithms for recommending cell phone games
[Jannach, Hegelich 09]
- Involved 150,000 users on a commercial mobile internet portal
- Comparison of recommender methods
- Random assignment of users to a specific method



online evaluation: characteristics of methods

who is the subject that is in the focus of research?	online customers, students, historical online sessions, computers, ...
what research methods are applied?	experiments, quasi-experiments, non-experimental research
in which setting does the research take place?	lab, real-world scenarios

Evaluation settings

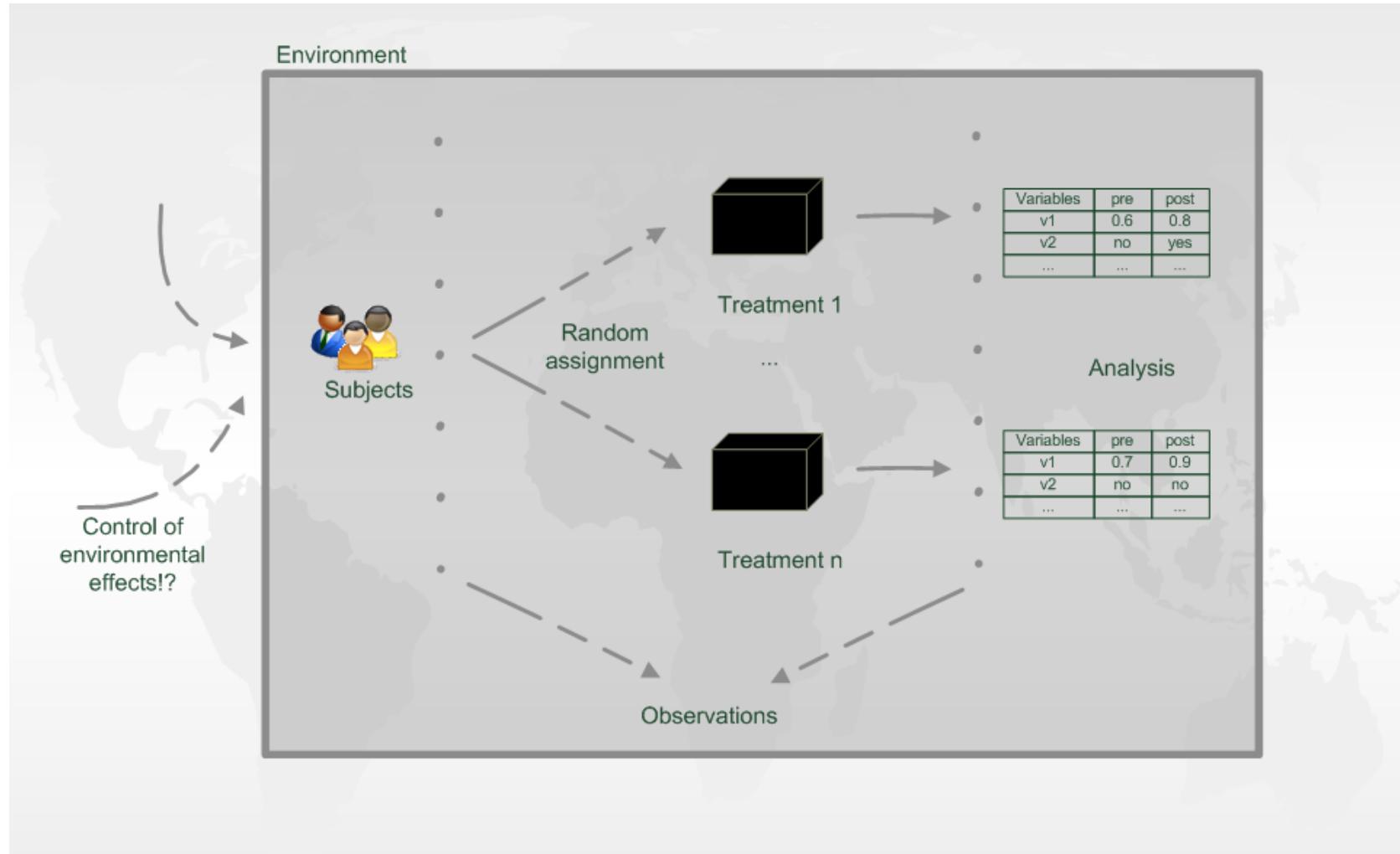
Lab studies

- Expressly created for the purpose of the study
- Extraneous variables can be controlled more easily by selecting study participants
- ... who should behave as they would in a real-world environment
- ... but doubts may exist about participants motivated by money, prizes or social pressure

Field studies

- Conducted in a preexisting real-world environment
- Users are intrinsically motivated to use a system

experiment design: A/B testing



back to game recommendation: experimental design



- Hypotheses on personalized vs. non-personalized recommendation techniques and their potential to
 - Increase conversion rate (i.e. the share of users who become buyers)
 - Stimulate additional purchases (i.e. increase the average shopping basket size)
- 155,000 visitors to site
 - split into 6 groups
 - ensure that customer profiles contained enough information (ratings) for all variants to make a recommendation
 - groups were chosen to represent similar customer segments
- catalog of 1,000 games
- five-point ratings scale ranging from -2 to +2 to rate items
 - due to the low number of explicit ratings, a click on the “details” link for a game was interpreted as an implicit “0” rating and a purchase as a “1” rating

Non-experimental research

- Quasi-experiments
 - Lack random assignments of units to different treatments
- Non-experimental / observational research
 - Surveys / Questionnaires
 - Longitudinal research
 - Observations over long period of time
 - E.g. customer life-time value, returning customers
 - Case studies
 - Focus on answering research questions about how and why
 - E.g. answer questions like: *How recommendation technology contributed to Amazon.com's becomes the world's largest book retailer?*
 - Focus group
 - Interviews
 - Think aloud protocols

analysis of results in general

- not different from other ML tasks
- are observed differences statistically meaningful or due to chance?
 - standard procedure for testing the statistical significance of two deviating metrics is the pairwise analysis of variance (ANOVA)
- ... and are they of practical importance?
 - statistically significant doesn't mean important
 - what is the value to the organization?

- introduction
- collaborative filtering
- evaluation
- data
 - types of ratings
 - sparsity
- more algorithms
- challenges

ratings: explicit vs implicit

- **explicit**
 - typical choices:
 - 1 to 5, 1 to 7 Likert response scales
 - probably the most precise ratings
 - ... possibly multidimensional
 - e.g. ratings for actors and sound as opposed to the movie
 - main challenge
 - users not always willing to rate many items
- **implicit**
 - user action interpreted as rating
 - e.g. access to content in social media
 - ... access to product's page and/or buying it
 - easy to collect transparently, without additional effort
 - main challenge
 - action doesn't necessarily have the same meaning as a rating
 - e.g. user might not like all the books he or she has bought; the user also might have bought a book for someone else

User\Item	Item 1	Item 2	Item 3	Item 4	Item N-2	Item N-1	Item N
User 1	1		1						
User 2				1				1	
User 3	1		1		1		1		
User 4									1
...									
...		1			1				
User M-2						1			
User M-1								1	
User M		1				1			

<https://github.com/CSKrishna/Recommender-Systems-for-Implicit-Feedback-datasets>

the truth about ground truth

[in RS, at least]

- what is the meaning of an unknown?

Offline experimentation	Online experimentation
Ratings, transactions	Ratings, feedback
Historic session (not all recommended items are rated)	Live interaction (all recommended items are rated)
Ratings of unrated items unknown, but interpreted as “bad” (default assumption, user tend to rate only good items)	“Good/bad” ratings of not recommended items are unknown
If default assumption does not hold: True positives may be too small False negatives may be too small	False/true negatives cannot be determined
Precision may increase Recall may vary	Precision ok Recall questionable

data is sparse

[in RS, at least]

- natural datasets include historical interaction records of real users
 - what proportion of products from the Amazon catalog does a regular customer buy?
- sparsity can be measured
 - Sparsity = $1 - |R|/|I| \cdot |U|$
 - R = ratings
 - I = items
 - U = users

User\Item	Item 1	Item 2	Item 3	Item 4	Item N-2	Item N-1	Item N
User 1	1		1						
User 2				1				1	
User 3	1		1		1		1		
User 4									1
...									
...		1			1				
User M-2						1			
User M-1								1	
User M		1				1			

<https://github.com/CSKrishna/Recommender-Systems-for-Implicit-Feedback-datasets>

the problems with sparsity

- how many items in common are two users expected to have?
 - so, how likely are patterns with large statistical support?
- additional (very interesting!) problem
 - cold start problem
 - how to recommend new items?
 - what to recommend to new users?
 - some (simple) approaches
 - ask/force users to rate a set of items
 - in the beginning, use method not based on ratings
 - ... then CF method
 - default voting
 - assign default values to items that only one of the two users to be compared has rated
 - more complex algorithms exist

User\Item	Item 1	Item 2	Item 3	Item 4	Item N-2	Item N-1	Item N
User 1	1		1						
User 2				1				1	
User 3	1		1		1		1		
User 4									1
...									
...		1			1				
User M-2						1			
User M-1								1	
User M		1				1			

<https://github.com/CSKrishna/Recommender-Systems-for-Implicit-Feedback-datasets>

... also for evaluation

Nr.	UserID	MovieID	Rating (r_i)	Prediction (p_i)	$ p_i - r_i $	$(p_i - r_i)^2$
1	1	134	5	4.5	0.5	0.25
2	1	238	4	5	1	1
3	1	312	5	5	0	0
4	2	134	3	5	2	4
5	2	767	5	4.5	0.5	0.25
6	3	68	4	4.1	0.1	0.01
7	3	212	4	3.9	0.1	0.01
8	3	238	3	3	0	0
9	4	68	4	4.2	0.2	0.04
10	4	112	5	4.8	0.2	0.04
					4.6	5.6

MAE = 0.29
RMSE = 0.42

MAE = 0.46
RMSE = 0.75

- introduction
- collaborative filtering
- evaluation
- data
- more algorithms
 - more about CF approaches
 - content-based
 - knowledge-based
 - hybrid approaches
- challenges

Memory-based and model-based approaches

- User-based CF is a memory-based approach
 - the rating matrix is directly used to find neighbors / make predictions
 - does not scale for most real-world scenarios
- Model-based approaches
 - based on an offline pre-processing or "model-learning" phase
 - at run-time, only the learned model is used to make predictions
 - models are updated / re-trained periodically

More model-based approaches

- Matrix factorization techniques, statistics
 - singular value decomposition, principal component analysis
- Association rule mining
 - compare: shopping basket analysis
- Probabilistic models
 - clustering models, Bayesian networks, probabilistic Latent Semantic Analysis
- Various other machine learning approaches

Matrix factorization

- (Golub and Kahan 1965) a given matrix M can be decomposed into a product of three matrices as follows

$$M = U \times \Sigma \times V^T$$

- where U and V are called *left* and *right singular vectors* and the values of the diagonal of Σ are called the *singular values*
- full matrix can be approximated by observing only the most important features
 - those with the largest singular values

Example for SVD-based recommendation

- SVD: $M_k = U_k \times \Sigma_k \times V_k^T$

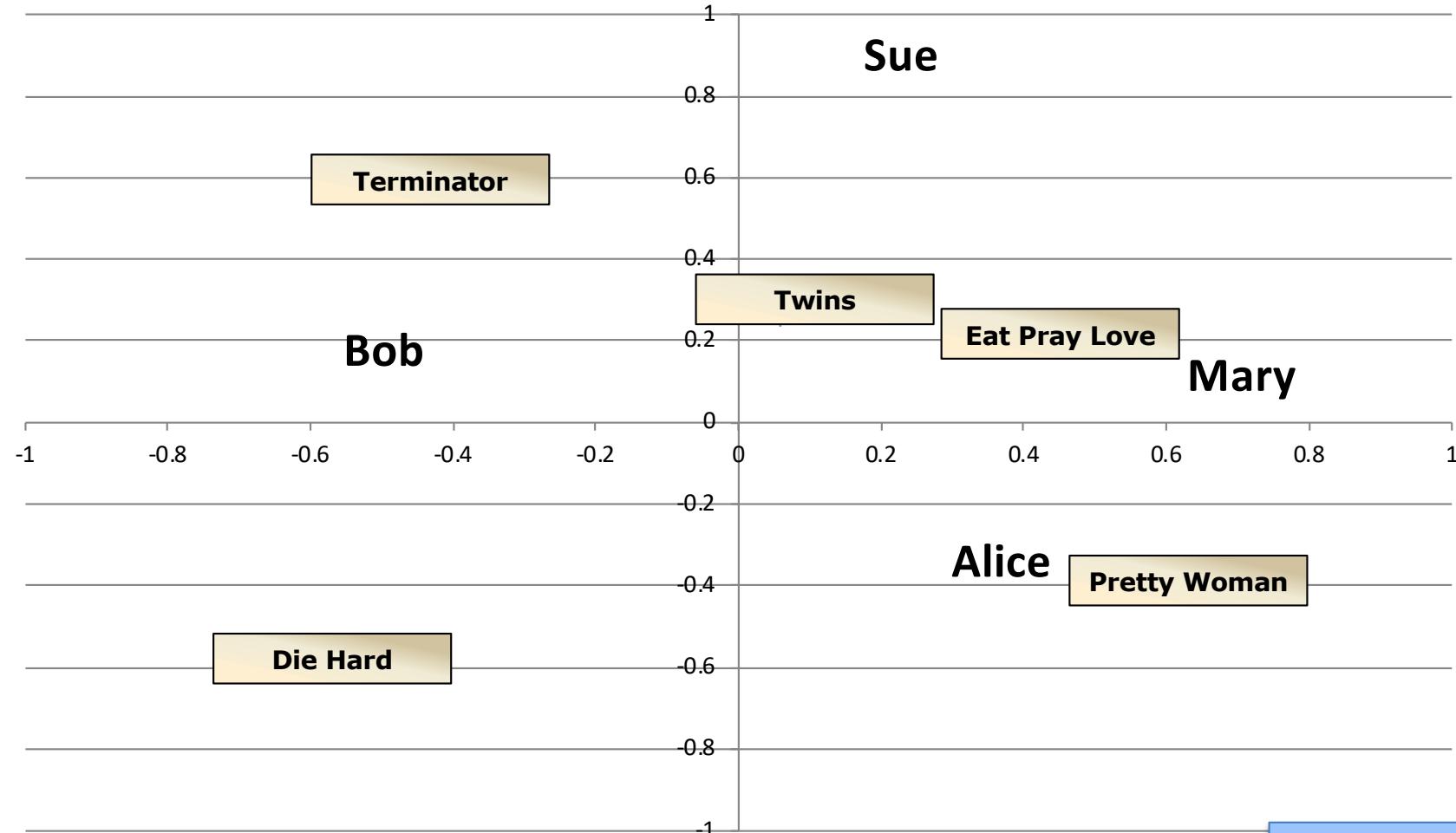
U_k	Dim1	Dim2
Alice	0.47	-0.30
Bob	-0.44	0.23
Mary	0.70	-0.06
Sue	0.31	0.93

V_k^T	Terminator	Die Hard	Twins	Eat Pray Love	Pretty Woman
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

- Prediction: $\hat{r}_{ui} = \bar{r}_u + U_k(Alice) \times \Sigma_k \times V_k^T(EPL)$
 $= 3 + 0.84 = 3.84$

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

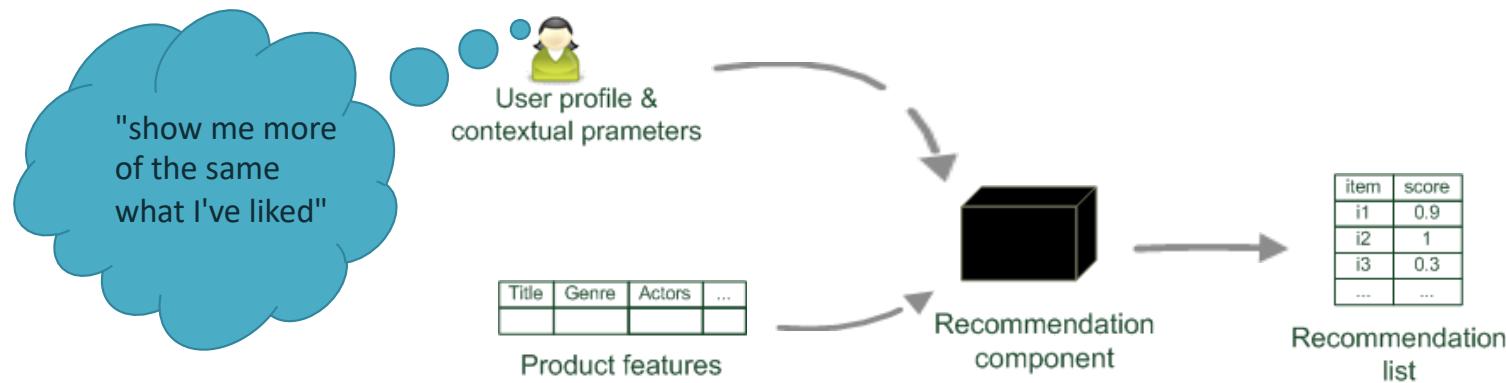
projection of U and V^T in the 2 dimensional space (U_2, V_2^T)



remember PCA?

Content-based recommendation

- While CF – methods do not require any information about the items,
 - it might be reasonable to exploit such information; and
 - recommend fantasy novels to people who liked fantasy novels in the past
- What do we need:
 - some information about the available items such as the genre ("content")
 - some sort of *user profile* describing what the user likes (the preferences)
- The task:
 - learn user preferences
 - locate/recommend items that are "similar" to the user preferences



what is the "content"?

- often, a combination of attributes and (semi-)free text
 - e.g. book recommendation

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism



- recommendation approach
 - related to NLP and document classification
 - ... out of the scope of this course

knowledge-based recommendations

- users want to define their requirements explicitly
 - "the accomodations should be pet-friendly"
- time span plays an important role
 - e.g. five-year-old ratings for computers are hardly useful
 - ... or user lifestyle or family situation changes
- items with low number of available ratings
 - typically

Search

Destination/property name:

Check-in date:

Check-out date:

2 adults - 0 children - 1 room

Entire homes & apartments
 I'm travelling for work

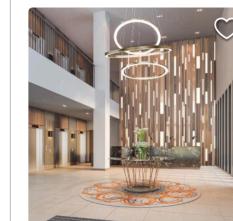
Filter by:

Health & safety
 Properties that take health & safety measures 377

Popular filters
 Hotels 129
 Indoor pool 13
 Hot tub/Jacuzzi 16
 Holiday homes 98
 5 stars 11

Manchester: 586 properties found

Commission paid and other benefits may affect an accommodation's ranking. [Find out more](#)



Clayton Hotel Manchester City C

[Manchester City Centre, Manchester](#) · [Show on map](#)
400 m from centre

[Travel Sustainable property](#)
In a prime location in the centre of Manchester, Clayton City Centre provides air-conditioned rooms, a fitness



Motel One Manchester-Piccadilly

[Manchester City Centre, Manchester](#) · [Show on map](#)
500 m from centre

Motel One Manchester-Piccadilly is located a 5-min walk from Manchester Piccadilly train station, offering a central WiFi and use of on-site bar One Lounge.



The Midland ★★★★

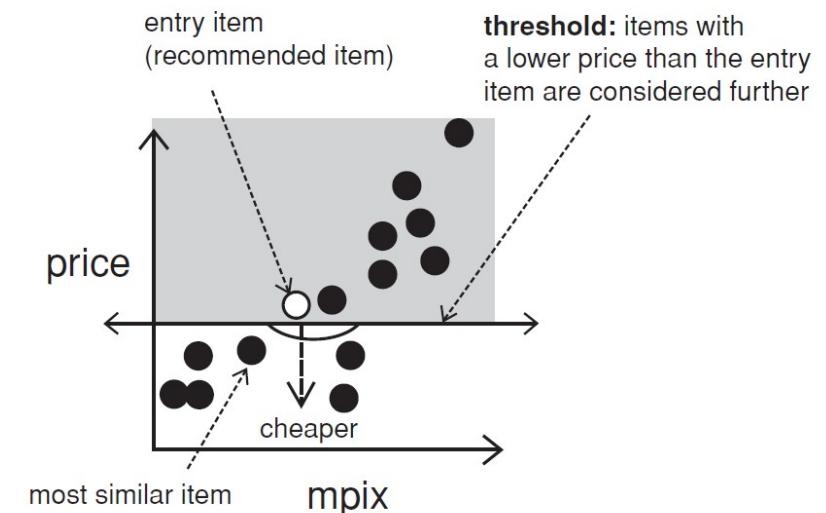
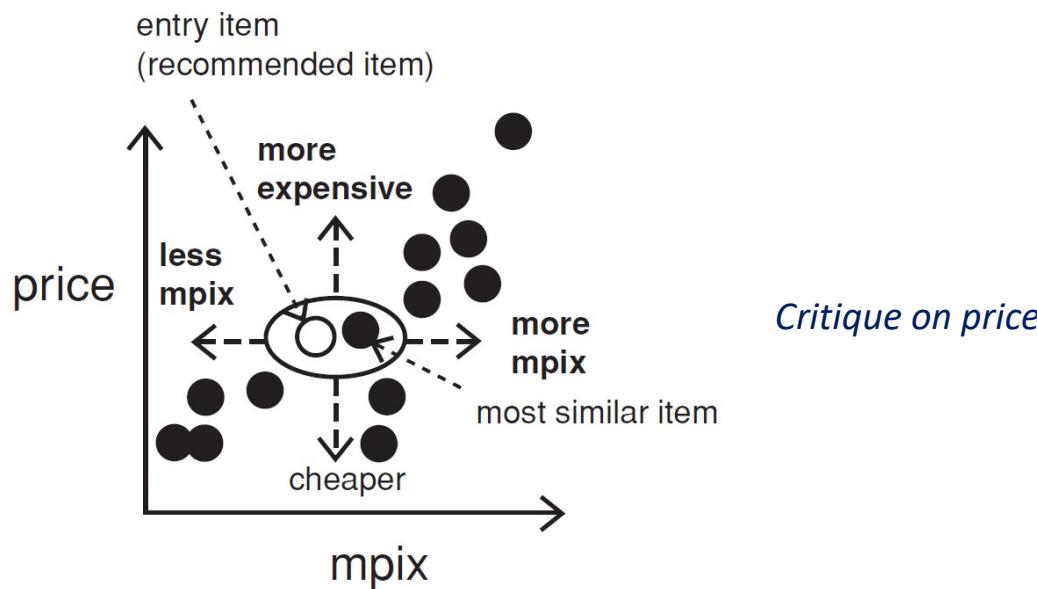
[Manchester City Centre, Manchester](#) · [Show on map](#)
700 m from centre

knowledge-based RS

- Constraint-based
 - based on explicitly defined set of recommendation rules
 - fulfill recommendation rules
- Case-based
 - based on different types of similarity measures
 - retrieve items that are similar to specified requirements
- Both approaches are similar in their **conversational** recommendation process
 - users specify the requirements
 - systems try to identify solutions
 - if no solution can be found, users change requirements

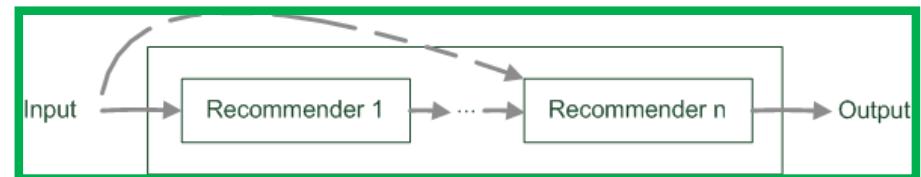
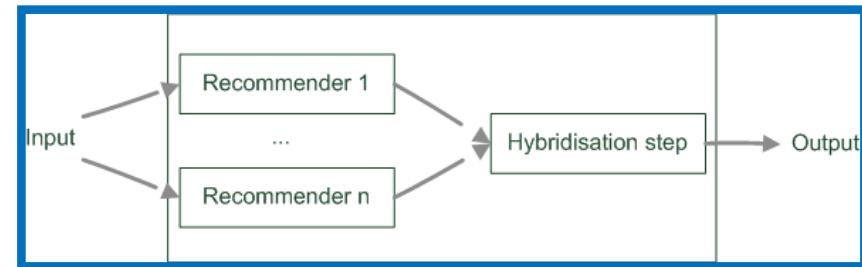
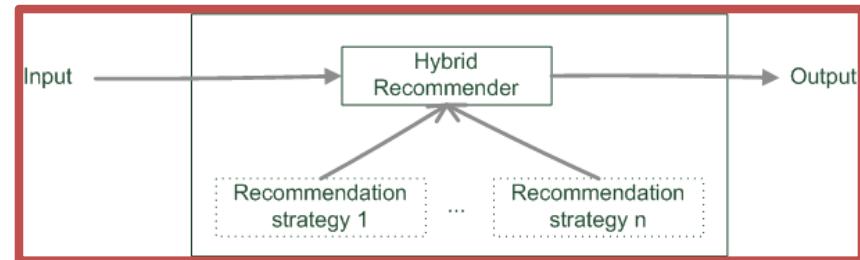
interaction with knowledge-based RS: critiquing

- user may not know exactly what they are seeking
- ... can specify their why current item is not satisfactory
 - e.g. price must be lower



hybrid recommender systems

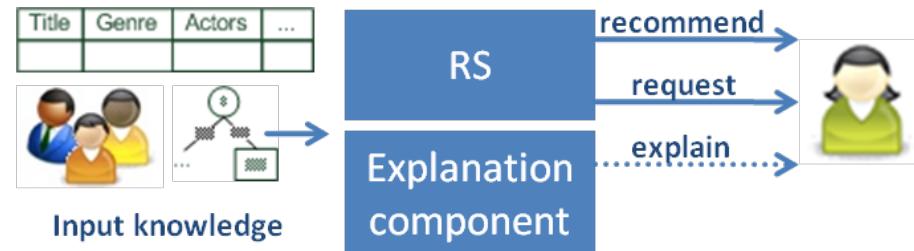
- think of the best salesperson you have met
 - probably combines ideas from the three approaches discussed
- hybridization
 - monolithic exploitation of different features
 - parallel
 - pipeline



- introduction
- collaborative filtering
- evaluation
- data
- more algorithms
- challenges
 - explaining recommendations
 - attacks
 - ubiquitous recommendations

explanation in recommender systems

- “The digital camera *Profishot* is a must-buy for you because . . .”
- two parties involved
 - organization interested in convincing user
 - user concerned about making the right choice(s)



attacks

- (monetary) value of being in recommendation lists
- attacks aim to
 - push some items
 - sabotage other items
 - simply sabotage the system
 - manipulation the "internet opinion"

example: profile injection

- e.g. memory-based CF with 1 neighbour
 - i.e. only opinion of most similar user will be used to make prediction

	Item1	Item2	Item3	Item4	...	Target	Similarity
Alice	5	3	4	1	...	?	
User1	3	1	2	5	...	5	-0.54
User2	4	3	3	3	...	2	0.68
User3	3	3	1	5	...	4	-0.72
User4	1	5	5	2	...	1	-0.02
Attack	5	3	4	3	...	5	0.87

← User2 most similar to Alice

← Attack most similar to Alice

Attack

ubiquitous RS: there's an app for it

- mobile applications have been a domain for recommendation
 - small display sizes and space limitations
 - naturally require personalized information
 - used “on the go”



context-aware recommendation

- (Ranganathan and Campbell, 2003) context
 - "any information about the circumstances, objects or conditions surrounding a user that is considered relevant to the interaction between the user and the ubiquitous computing environment"
- (Shilit et al., 1994) most important aspects of context as
 - where you are
 - who you are with
 - what resources are nearby

research questions in ubiquitous domains

- goals
 - serendipitous recommendations vs proximity?
- role of contextual parameters, such as location
 - another preference
 - a requirement that is always strictly enforced, or
 - something in between?
- modality of interaction, for users "on the go"
 - pushing information is useful to draw recipients' attention
 - ... but may be invasive

application domains

- M-Commerce
 - m-commerce refers to monetary transactions that are conducted via wireless networks.
- Tourism and visitor guides
 - travelers have specific information needs, makes this domain a natural choice for mobile information systems.
- Cultural heritage and museum guides
 - mobile guides for archeological sites or museums providing multimedia services.
- Home computing and entertainment
 - users are able to personally configure and adapt smart devices in their environment based on their preferences and on specific situations.

discussion & summary

- problem of recommendation
- collaborative filtering approaches
- ... and other algorithms
- evaluation is key! (once more...)
 - metrics
 - ... use of data for estimating their value
- issues of the data used in RS
- ... other challenges

ensemble learning

Carlos Soares

(based on materials kindly provided
by João Mendes Moreira)

plan & goals

- introduction
- categories of methods
- popular methods
- issues
- understand the basic principles of ensemble learning
- understand the intuition and high-level algorithm of some of the most common ensemble methods

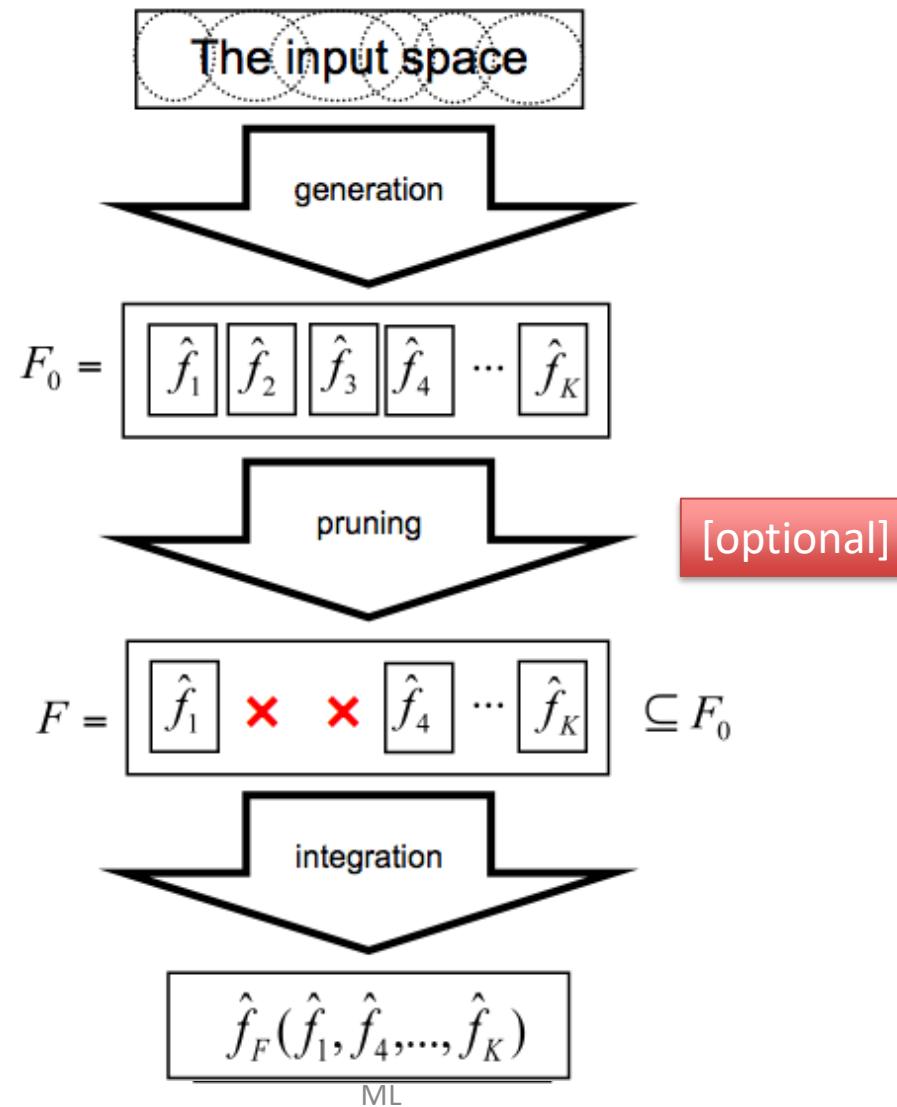
definition

- multiple models
 - **base models**
- ... each of them obtained by applying a learning process to a given problem
 - e.g. same algorithm applied to different samples of the data
- ... combined to make a single prediction
 - e.g. in classification, each model makes a prediction
 - ... then combined to obtain the final prediction of the ensemble

intuition

- aggregation of multiple learned models with the goal of improving model quality
 - e.g. expert panel in a human decision-making process
 - ... or the popular concept of “the wisdom of the crowds”

ensemble learning process



discussion

why should ensemble methods work?
[even better, when...?]

what's the catch?

pros & cons

+

- accuracy
 - majority compensates for individual errors
- diversity is key
 - individual models specialize in different areas of the data space
 - how?
 - ... but must be reasonably accurate
 - ... and by “reasonable” we mean...?

-

- complexity
 - understanding the global model
 - explaining decisions
 - computational
- remember Occam’s Razor
 - simplicity leads to greater accuracy
 - identifying the best model requires identifying the proper “model complexity”

- introduction
- categories of methods
 - homogeneous
- popular methods
- issues

ensembles methods for...

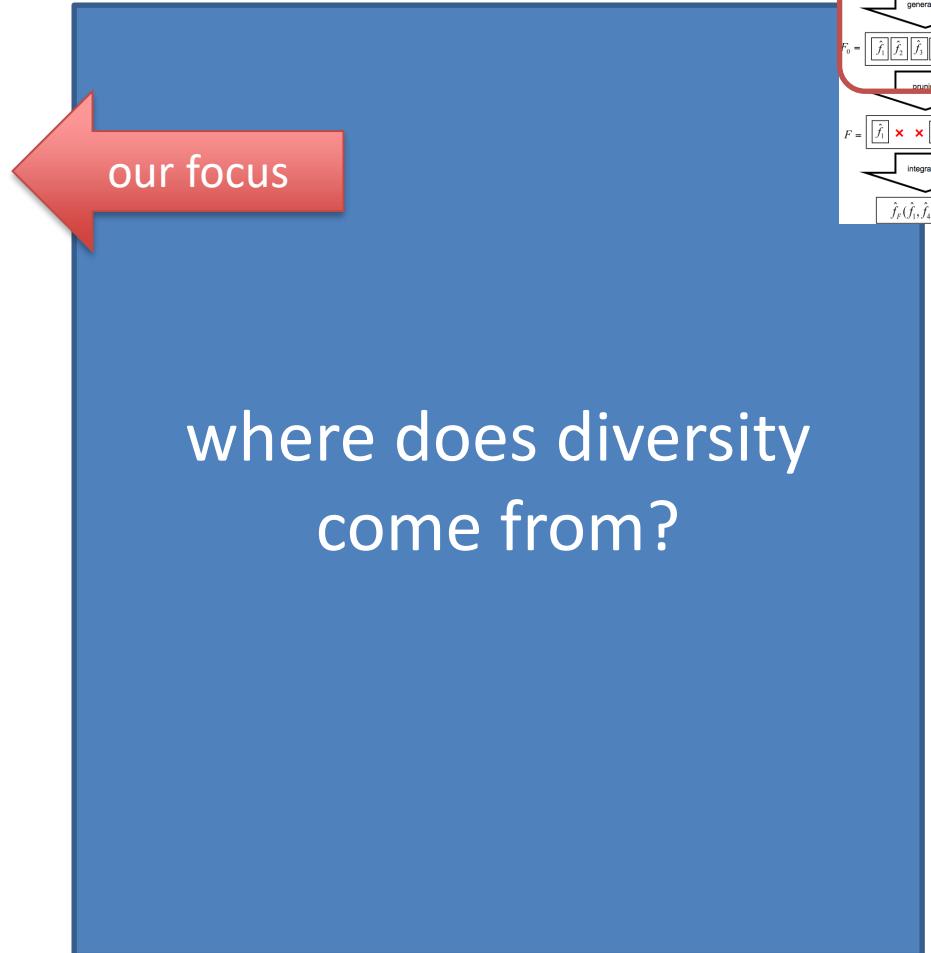
- classification
- regression
- clustering
 - aka consensual clustering
- label ranking
- ...
 - anything, really



types of ensembles: how to generate models

- homogeneous
 - single induction algorithm

- heterogeneous
 - multiple induction algorithms



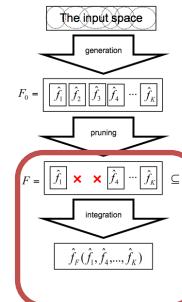
types of ensembles: how to combine models

regression

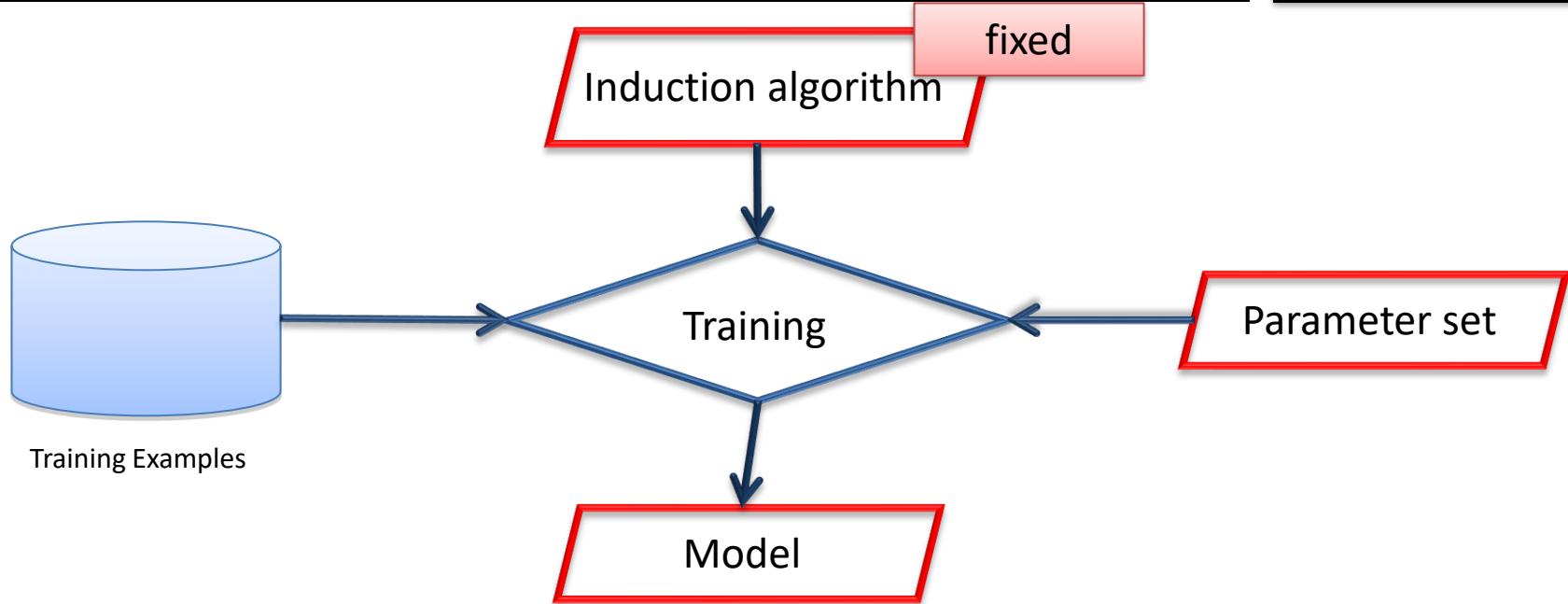
- **average**
- **weighted average**
- sum
- weighted sum
- product
- maximum
- minimum
- median

classification

- **majority voting**
- **weighted majority voting**
- borda count
 - base models rank candidates in order of preference
 - e.g. remember scoring?
 - points assigned to each position
 - prediction is class with more points



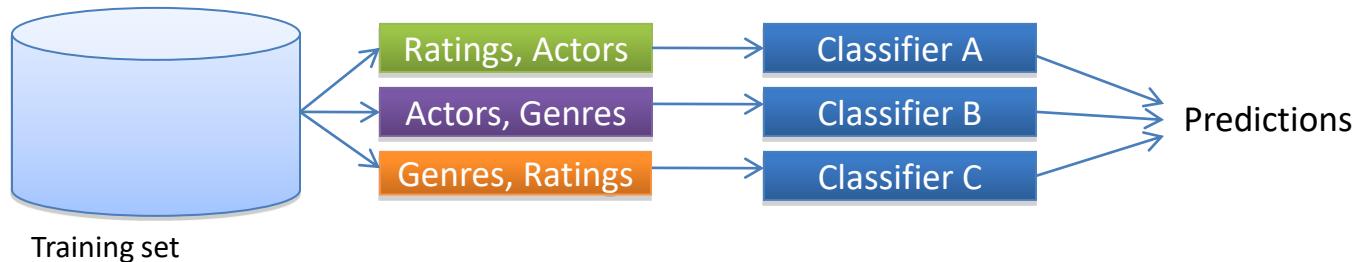
homogeneous ensembles: how to generate different models?



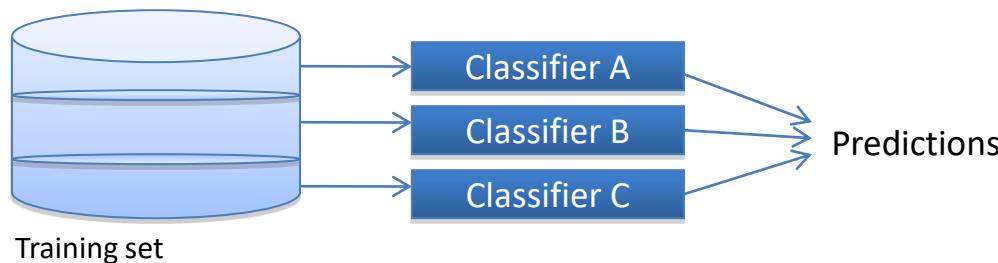
- Data manipulation
 - training set
- Modeling process manipulation
 - induction algorithm
 - parameter set
 - model
 - uncommon

data manipulation

Manipulating the input features

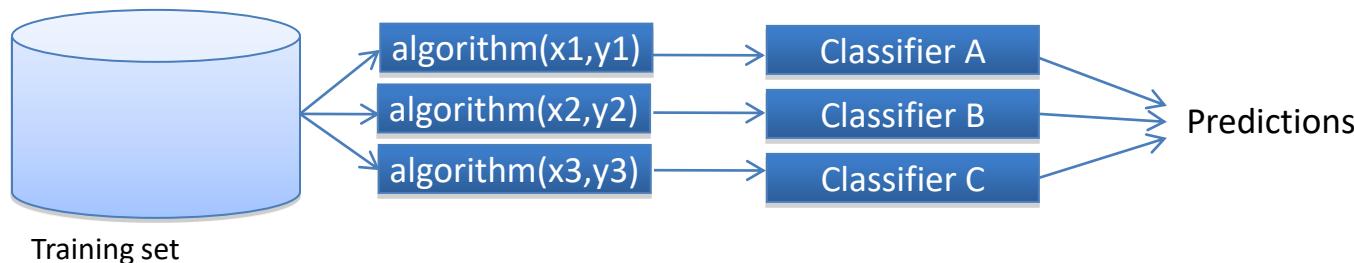


Sub-sampling from the training set

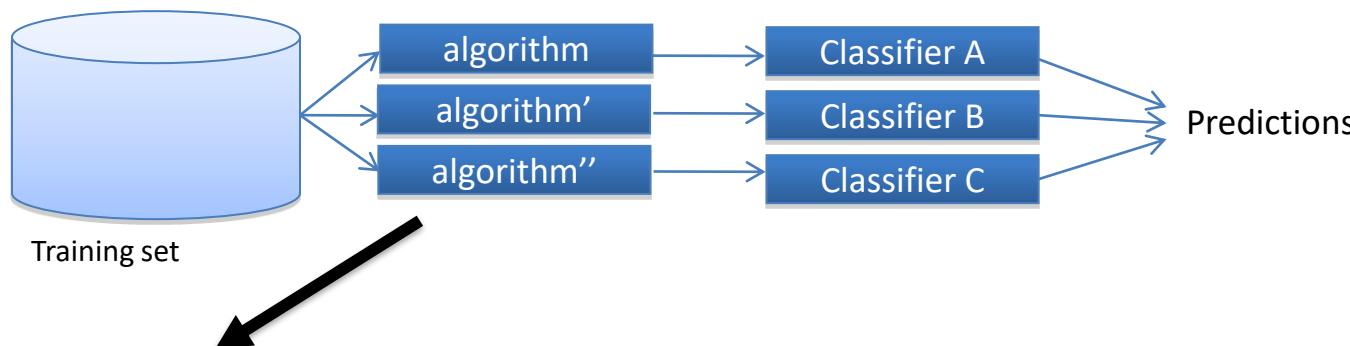


modeling process manipulation

Manipulating the parameter sets



Manipulating the induction algorithm

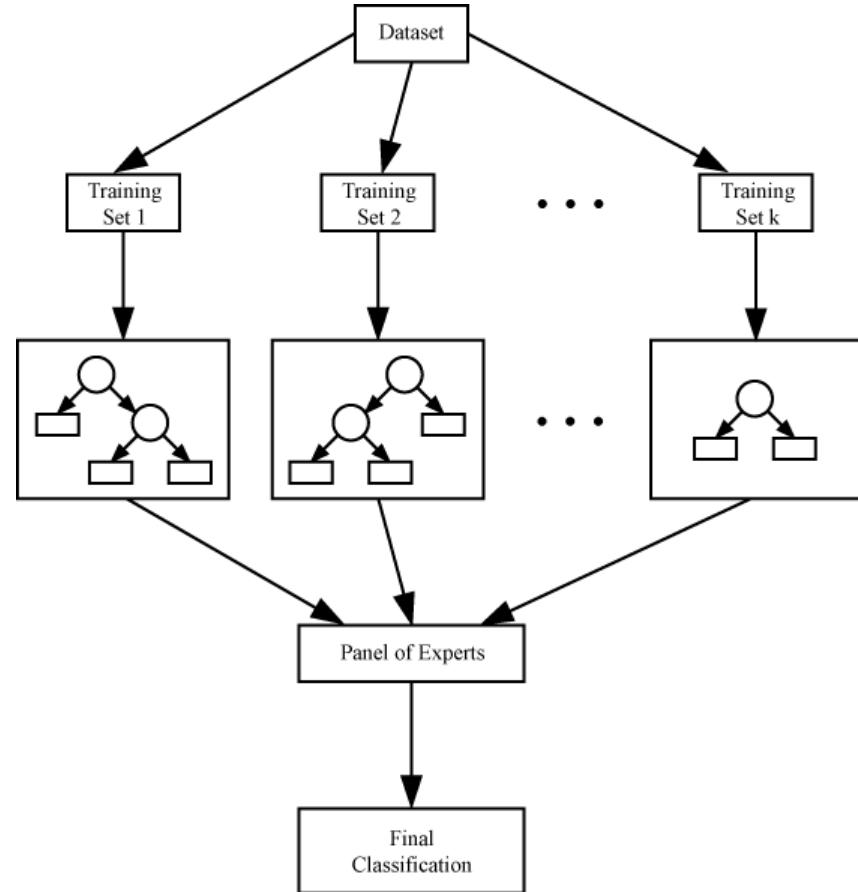


still homogeneous: *algorithm'* and *algorithm''* are variations of *algorithm*

- introduction
- categories of methods
- popular methods
 - bagging
 - boosting
 - random forest
 - negative correlation
- issues

bagging: Bootstrap AGGREGatING

- diagnosis analogy
 - diagnosis based on the majority vote of multiple doctors
 - trained in slightly different contexts
- training
 - given a set D of d tuples
 - at each iteration i
 - training set D_i of d tuples is sampled with replacement from D
 - i.e. bootstrap
 - model M_i is learned for training set D_i
- prediction
 - given an observation X
 - for each classifier M_i
 - make a prediction
 - an aggregation of the predictions is the prediction of the bagged model M^* for X



http://en.wikibooks.org/wiki/File:DTE_Bagging.png

bagging

- accuracy
 - often significantly better than a single classifier derived from D
 - robust to noise
- ... if classifier is unstable!
 - unstable means a small change to the training data may lead to major decision changes
 - decision trees
 - neural networks

boosting

- training
 - equal weights are assigned to each training example
 - learn model M_1
 - learn additional $k-1$ models
 - give more weight to the examples that were incorrectly predicted by M_i
 - learn model M_{i+1}
- prediction
 - given an observation X
 - for each classifier M_i
 - make a prediction
 - an aggregation of the predictions is the prediction of the bagged model M^* for X
 - the weight of each classifier's vote is a function of its accuracy

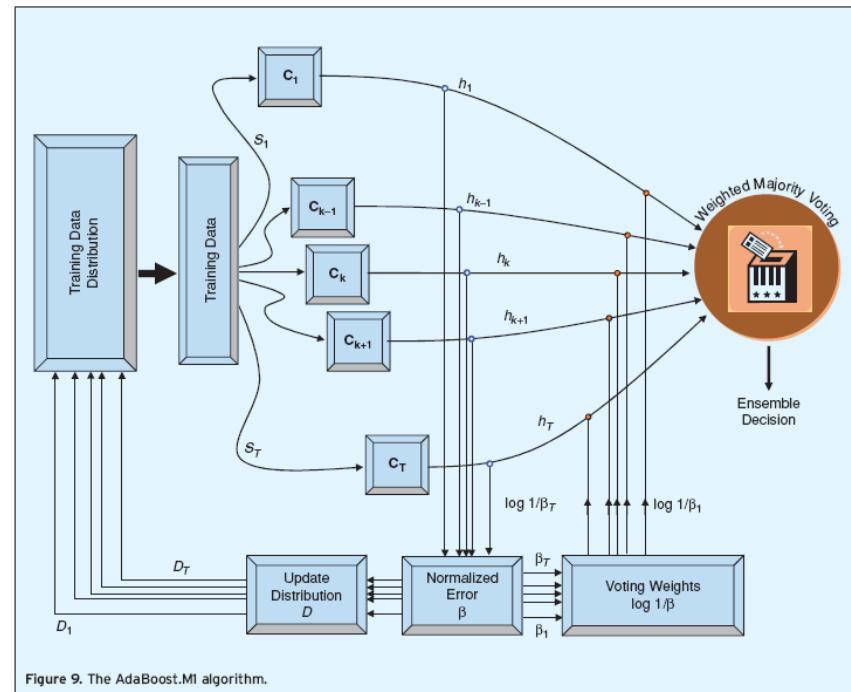


Figure 9. The AdaBoost.M1 algorithm.

boosting: discussion

- boosting vs. bagging
 - differences
 - independent sampling vs. error-dependent sampling
 - uniform aggregation vs. weighted aggregation
- ... SO
 - boosting tends to achieve greater accuracy
 - ... but it also risks overfitting the model to misclassified data

random forest

- **training**
 - learn k models
 - ... with changed algorithm
 - at each split
 - randomly select a subset of the original features during the process of tree generation
- **prediction**
 - given an observation X
 - for each classifier M_i
 - make a prediction
 - an aggregation of the predictions is the prediction of the bagged model M^* for X

random forest: discussion

- RF vs adaboost
 - comparable in accuracy
 - more robust to errors and
 - ... outliers
- ... vs bagging and adaboost
 - RF is insensitive to the number of attributes selected for consideration at each split and
 - faster

negative correlation learning

- **training**
 - learn k models
 - ... with changed algorithm
 - trained to minimize the error function of the ensemble
 - i.e., it adds to the error function a penalty term with the average error of the models already trained
- **prediction**
 - given an observation X
 - for each classifier M_i
 - make a prediction
 - an aggregation of the predictions is the prediction of the bagged model M^* for X

negative correlation learning: discussion

- only regression
 - algorithms that try to minimize/maximize a given objective function
 - e.g., neural networks, support vector regression
- models negatively correlated with the averaged error of the previously generated models

popular ensemble methods: summary

- bagging
 - base models: train algorithm on different bootstrap samples
 - prediction: average/majority
 - task: classification and regression
- boosting
 - base models: sequence of training processes, with more weight given to instances incorrectly classified by previous model
 - prediction: weighted vote
 - task: classification
- random forest
 - base models: train algorithm on different samples of attributes
 - prediction: average/majority
 - task: classification and regression
- negative correlation learning
 - base models: sequence of training processes, with new models forced to be more negatively correlated with the existing ones
 - prediction: average
 - task: regression

- introduction
- categories of methods
- popular methods
- issues

characteristics of the base models: classification

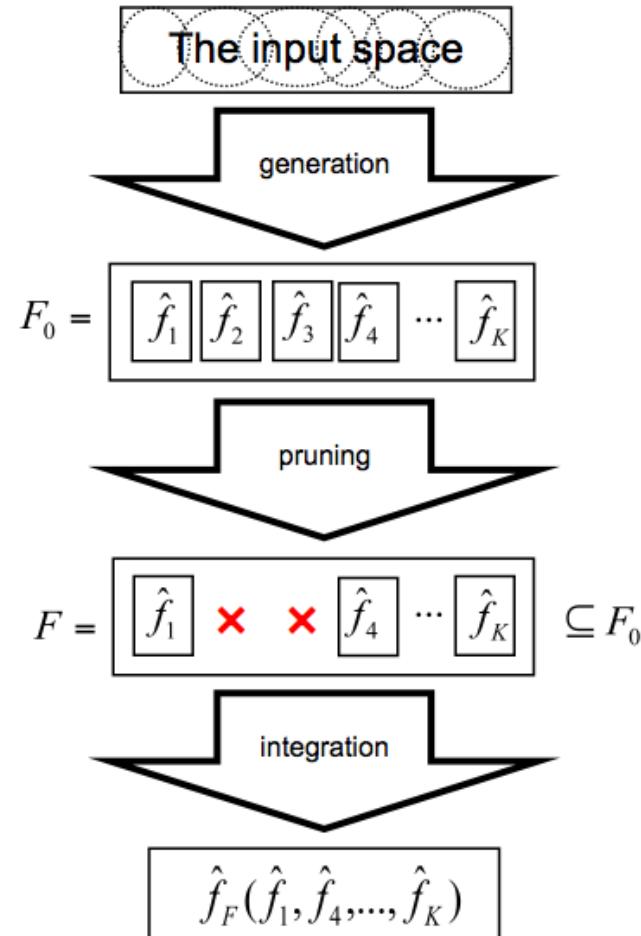
- base classifiers should be as accurate as possible and
 - although there is “the strength of weak classifiers”
 - R.E. Schapire. 1990. The Strength of Weak Learnability. *Mach. Learn.* 5, 2 (July 1990), 197-227
- having diverse errors
 - Brown, G. & Kuncheva, L., “Good” and “Bad” Diversity in Majority Vote Ensembles, *Multiple Classifier Systems, Springer*, **2010**, 5997, 124-133

characteristics of the base models: regression

- more amenable to theoretical analysis
 - the error of an ensemble \hat{f}_F with K base learners in relation to the true values given by f is:
 - $E(\hat{f}_F - f)^2 = \overline{\text{bias}}^2 + \frac{1}{K} \times \overline{\text{var}} + \left(1 - \frac{1}{K}\right) \times \overline{\text{covar}}$
 - ... assuming the integration function is the average
- the goal is to minimize $E(\hat{f}_F - f)^2$, so
 - the average bias of the base learners should be as small as possible
 - i.e. the base learners should be as accurate (on average) as possible
 - the average variance of the base learners should be as small as possible
 - i.e. the base learners should be as robust to small changes on the training data (on average) as possible
 - the average covariance of the base learners should be as low as possible
 - i.e. the base learners should have negative correlation

summary

- combination of multiple models
 - majority compensates for individual errors
- individual models specialize in different areas of the data space
 - diversity is key
- today
 - focused on homogeneous
 - but essentially applicable to heterogeneous ensembles



Introductory References

- *'Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations'*, Ian H. Witten and Eibe Frank, 1999
- *'Data Mining: Practical Machine Learning Tools and Techniques second edition'*, Ian H. Witten and Eibe Frank, 2005
- *Todd Holloway, 2008, "Ensemble Learning Better Predictions Through Diversity"*, power point presentation
- *Leandro M. Almeida, "Sistemas Baseados em Comitês de Classificadores"*
- *Cong Li, 2009, "Machine Learning Basics 3. Ensemble Learning"*
- R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, Quarter 2006.
- João Mendes-Moreira, Carlos Soares, Alípio Jorge, Jorge Freire de Sousa, "Ensemble approaches for regression: a survey", *ACM Computing surveys*, 45(1), article 10, 2012.

Core References

- Wolpert, D. H., Stacked generalization, *Neural Networks*, **1992**, 5, 241-259
- Breiman, L., Bagging predictors, *Machine Learning*, **1996**, 26, 123-140
- Freund, Y. & Schapire, R., Experiments with a new boosting algorithm, *International Conference on Machine Learning*, **1996**, 148-156
- Breiman, L., Random forests, *Machine Learning*, **2001**, 45, 5-32
- Liu, Y. & Yao, X., Ensemble learning via negative correlation, *Neural Networks*, **1999**, 12, 1399-1404
- Rodríguez, J. J.; Kuncheva, L. I. & Alonso, C. J., Rotation forest: a new classifier ensemble, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2006**, 28, 1619-1630

Outlier Detection

Rita P. Ribeiro

Machine Learning - 2021/2022



Summary

1. Basic Concepts

Definition of Outlier

Application Domains

Challenges

Key Aspects

2. Outlier Detection Approaches

Unsupervised Learning Techniques

Semi-supervised Learning Techniques

Advanced Topics

3. Summary

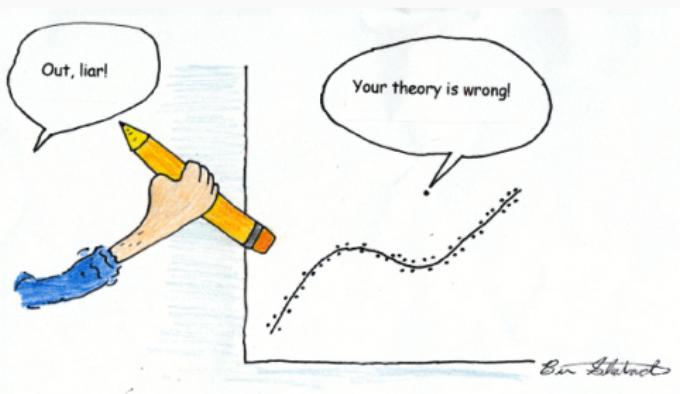
Basic Concepts

Motivation

- Most of data mining tasks focus on creating a model of the “normal” patterns in the data, extracting knowledge from what is common (e.g. frequent patterns).
- Still, rare patterns can also give us some import insights about data.
- Depending on the goal, those insights can be even more interesting/critical than the “normal” patterns.

What is an Outlier?

- “An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism” (Hawkins, 1980)



What is an Outlier? (cont.)

- Outliers represent patterns in data that do not conform to a well defined notion of normal.
- Initially, outliers were considered errors and their identification had data cleaning purposes.
- However, they can represent truthful deviation of data.
- For some applications, they represent critical information, which can trigger preventive or corrective actions.



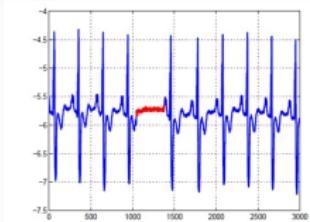
Outliers and Anomalies

- Outlier and Anomaly detection are roughly related.
- Outliers can have a negative connotation being associated with data noise.
- Anomalies are often associated with unusual data that should be further investigated to identify the cause of occurrence.
- Anomaly can be considered as an outlier.
- But an outlier is not necessarily an anomaly.
- The following outlier detection application and methods involve outliers that can be seen as anomalies, i.e. meaningful outliers.

Where can Outliers occur?

Social Network Analysis

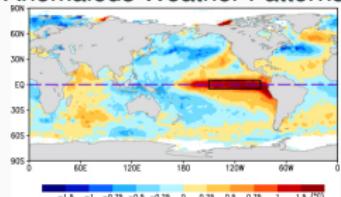
Medical Analysis



Financial Markets



Anomalous Weather Patterns



Fraud Detection



Event Detection in Text/Social Media



Applications of Outlier Detection

- Quality Control and Fault Detection Applications
 - Quality Control
 - Fault Detection and Systems Diagnosis
 - Structure Defect Detection
- Financial Applications
 - Credit Card Fraud
 - Insurance Claim Fraud
 - Stock Market Anomalies
 - Financial Interaction Networks
- Intrusion and Security Applications
 - Host-based Intrusions
 - Network Intrusion Detection
- Web Log Analytics
 - Web Log Anomalies

Applications of Outlier Detection (cont.)

- Market Basket Analysis
 - Outlier transactions in association patterns
- Medical Applications
 - Medical Sensor Diagnostics
 - Medical Imaging Diagnostics
- Text and Social Media Applications
 - Event Detection in Text and Social Media
 - Spam Email
 - Noisy and Spam Links
 - Anomalous Activity in Social Networks
- Earth Science Applications
 - Sea Surface Temperature Anomalies
 - Land Cover Anomalies
 - Harmful Algae Blooms

Challenges of Outlier Detection

- Define every possible “normal” behaviour is hard.
- The boundary between normal and a outlying behaviour is often not precise.
- There is no general outlier definition; it depends on the application domain.
- It is difficult to distinguish real meaningful outliers from simple random noise in data.
- The outlier behaviour may evolve with time.
- Malicious actions adapt themselves to appear as normal.
- Inherent lack of known labeled outliers for training/validation of models.

Key Aspects of Outlier Detection Problem

- Nature of Input Data
- Type of Outliers
- Intended Output
- Learning Task
- Performance Metrics

Nature of Input Data

- Each data instance has:
 - One attribute (univariate)
 - Multiple attributes (multivariate)
- Relationship among data instances:
 - None
 - Sequential / Temporal
 - Spatial
 - Spatio-temporal
 - Graph
- Dimensionality of data

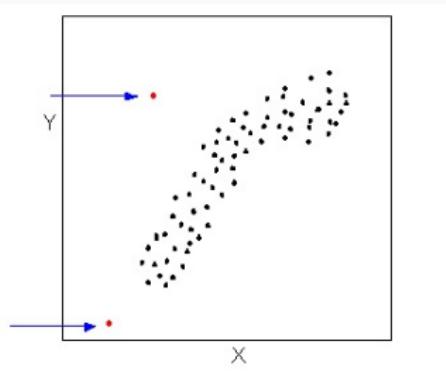
Types of Outliers

- Point (or Global) Outlier
- Contextual Outlier
- Collective Outlier

Types of Outliers (cont.)

Point Outlier

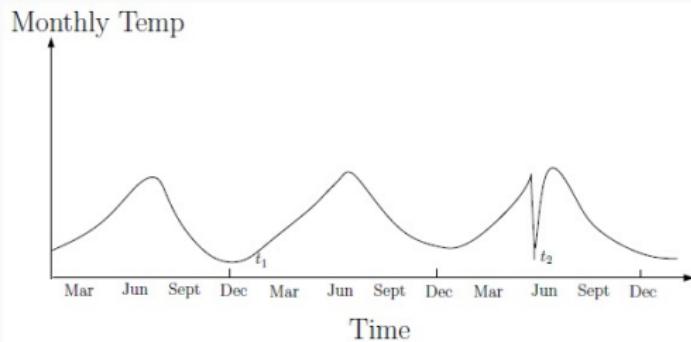
An instance that individually or in small groups is very different from the rest of the instances.



Types of Outliers (cont.)

Contextual Outlier

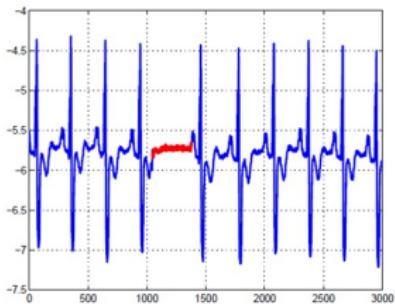
An instance that when considered within a context is very different from the rest of the instances.



Types of Outliers (cont.)

Collective Outlier

An instance that, even though individually may not be an outlier, inspected in conjunction with related instances and with respect to the entire data set is an outlier.



Intended Output

- Assign a **label/value**: identification normal or outlier instance.
- Assign a **score**: probability of being an outlier.
 - It allows the output to be ranked.
 - Requires the specification of a threshold.

Learning Task

Unsupervised Outlier Detection

- data set has no information on the behaviour of each instance;
- it assumes that instances with normal behaviour are far more frequent;
- most common case in real-life applications.

Semi-supervised Outlier Detection

- data set has a few instances of normal or outlier behaviour;
- some real-life applications, such as fault detection, provide such data.

Supervised Outlier Detection

- data set has instances of both normal and outlier behaviour;
- hard to obtain such data in real-life applications.

Inadequacy of Standard Performance Metrics

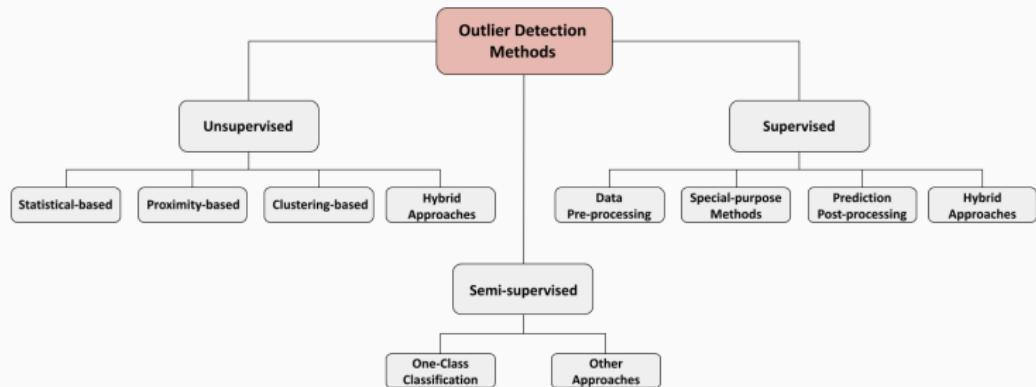
- Standard performance metrics (e.g. *accuracy*, *error rate*) assume that all instances are equally relevant for the model performance.
- These metrics would give a good performance estimation to a model that performs well on normal (frequent) cases and bad on outlier (rare) cases.

Credit Card Fraud Detection:

- data set D with only 1% of fraudulent transactions;
- model M predicts all transactions as non-fraudulent;
- M has an estimated accuracy of 99%;
- yet, all the fraudulent transactions were missed!

Outlier Detection Approaches

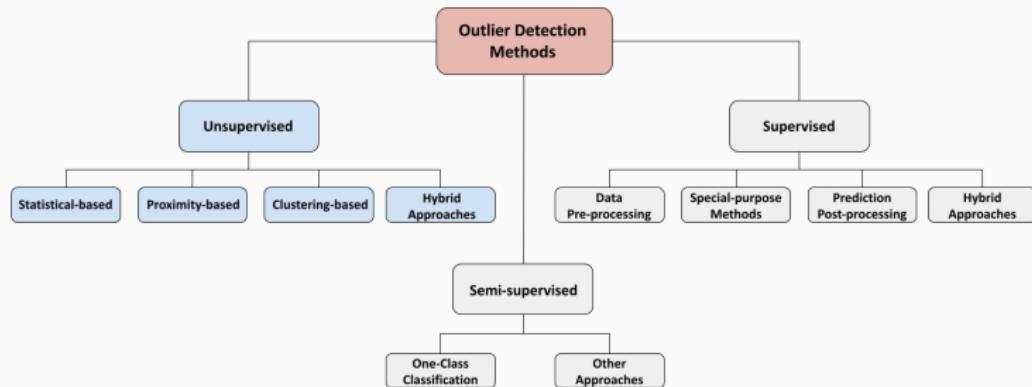
Taxonomy of Outlier Detection Methods



Outlier Detection Approaches

Unsupervised Learning Techniques

Taxonomy of Anomaly Detection Methods



Statistical-based Outlier Detection

Proposal

- All the points that satisfy a statistical discordance test for some statistical model are declared as outliers.

Advantages

- If the assumptions of the statistical model hold true, these techniques provide a justifiable solution for outlier detection.
- The outlier score is associated with a confidence interval.

Techniques

- Parametric
- Non-parametric

Statistical-based Outlier Detection: Parametric Techniques

Assume one of the known probability distribution functions.

- *Grubbs' Test* (Grubbs, 1950)

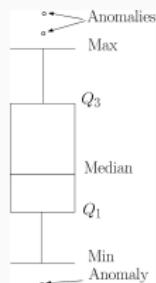
A statistical test used to detect outliers in a **univariate** data set assumed to come from a normally distributed population.

- *Boxplot* (Tukey, 1977)

It assumes a near-normal distribution of the values in a **univariate** data set, and identifies as outlier any value outside the interval

$$[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$$

where Q_1 (Q_3) is the 1st (3rd) quartile and IQR is the interquartile range.



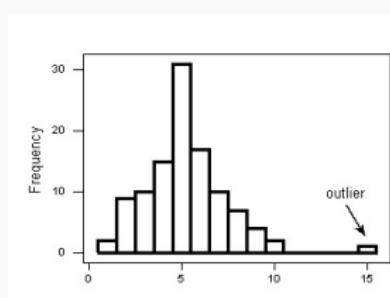
- *Mahalanobis distance* (Mahalanobis, 1936)
 - It assumes a multivariate normal distribution of data.
 - Incorporates dependencies between attributes by the covariance matrix.
 - Transforms a **multivariate** outlier detection task into a univariate outlier detection problem.
 - All the points with a large *Mahalanobis* distance are indicated as outliers.
- Mixture of parametric distributions
- etc.

Statistical-based Outlier Detection: Non-parametric Techniques

The probability distribution function is not assumed, but estimated from data.

- **Histograms**

- Used for both univariate and multivariate data. For the later, the attribute-wise histograms are constructed and an aggregated score is obtained.
- Hard to choose the appropriate bin size.



- **Kernel functions**

- Adopt a kernel density estimation to estimate the probability density distribution of the data.
- Outliers are in regions of low density.

Disadvantages

- The data does not always follows a statistical model.
- Choosing the best hypothesis test statistics is not straightforward.
- Capture interactions between attributes is not always possible.
- Estimating the parameters for some statistical models is hard.

Proximity-based Outlier Detection

Proposal

- Normal instances occur in dense neighbourhoods, while outliers occur far from their closest neighbours.

Advantages

- Purely data driven technique
- Does not make any assumptions regarding the underlying distribution of data.

Some Techniques

- Distance-based
- Density-based

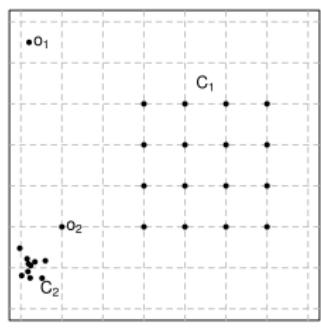
Proximity-based Outlier Detection: Distance-based Techniques

A case c is an outlier if less than k cases are within a distance λ of c
[Knorr and Ng, 1998]

- Outliers are points far away from other points, thus given a distance metric there should not be a lot of other points in their neighborhood.
- Define proper distance metric (e.g euclidean distance)
 - The notion of distance between cases with many variables may be distorted by different scales, different importance, different types (numerical, nominal)
- Define a “reasonable” neighborhood (λ)
- Define what is “a lot of other points” (k)

Proximity-based Outlier Detection: Distance-based Techniques (cont.)

- Major cost: for each point is calculated its distance to all the other points.
- The use of **global distance** measures poses difficulties in detecting outliers in data sets with different density regions.
- Example:



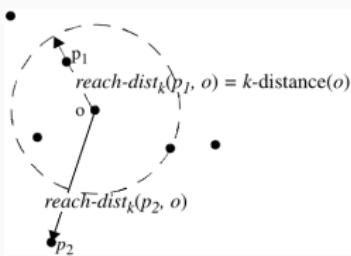
- o_1 and o_2 are outliers
- but, for the point o_2 to be identified as an outlier, all the points in C_1 would have to be identified as outliers too.

Proximity-based Outlier Detection: Density-based Techniques

- Concept of outliers should be **locally** inspected.
- Compare points to their local neighborhood, instead of the global data distribution
- The density around an outlier is significantly different from the density around its neighbours.
- Use the relative density of a point against its neighbours as the indicator of the degree of the point being an outlier.
- Outliers are points in lower local density areas with respect to the density of its local neighbourhood.

Proximity-based Outlier Detection: Density-based Techniques (cont.)

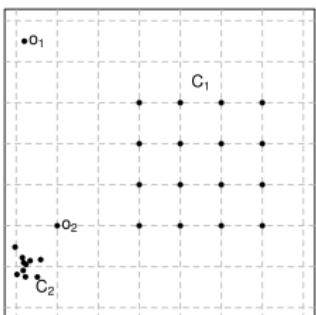
- LOF: Local Outlier Factor [Breunig et al., 2000]
 - k -distance: distance between p and its k -th nearest neighbour
 - k -distance neighborhood: all the points whose distance from p is not greater than the k -distance.
 - $reachability-distance$ of p with respect to o : the maximum between their k -distance and their actual distance.



- intuition: high values of reachability-distance between two given points indicates that they may not be in the same cluster

Proximity-based Outlier Detection: Density-based Techniques (cont.)

- LOF: Local Outlier Factor [Breunig et al., 2000] (cont.)
 - *local reachability-density* of a point is inversely proportional to the average reachability-distance of its k neighbourhood.
 - *LOF* assigns high values to the points that have a much lower *local reachability-density* in comparison to its k -neighbourhood.
 - Example:



- o_2 is assigned a higher LOF compared to the LOF values assigned to the points of C_1 and C_2
- This captures a local outlier whose local density is relatively low comparing to the local densities of its k -neighbourhood.

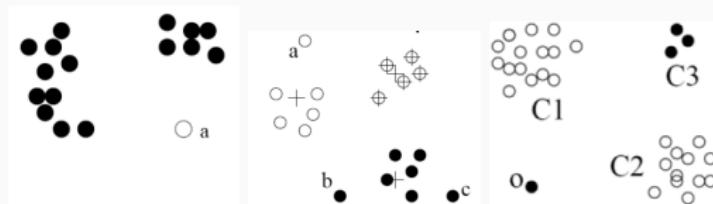
Disadvantages

- True outliers and noisy regions of low density may be hard to distinguish.
- These methods need to combine global and local analysis.
- In high dimensional data, the contrast in the distances is lost.
- Computational complexity of the test phase.

Clustering-based Outlier Detection

Proposal

- Normal instances belong to large and dense clusters, while outlier instances are instances that:
 - do not belong to any of the clusters;
 - are far from its closest cluster;
 - form very small or low density clusters.



Advantages

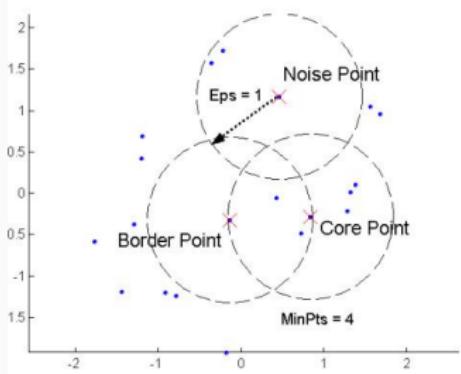
- Easily adaptable to on-line/incremental mode.
- Test phase is fast.

Clustering-based Outlier Detection: Techniques

- DBSCAN [Ester et al., 1996]

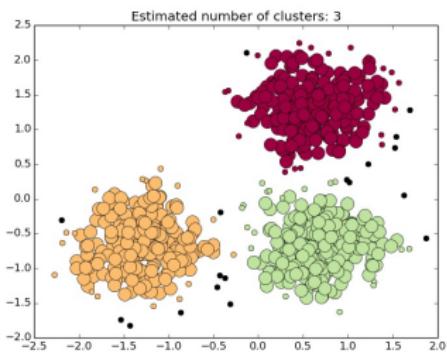
- Clustering method based on the notion of “density” of the points
- The density of a point is estimated by the number of points that are within a certain radius.
- Based on this idea, points can be classified as:

- *core points*: if the number of points within its radius are above a threshold
- *border points*: if the number of points within its radius are not above a threshold, but they are within a radius of a *core point*
- *noise points*: if do not have enough points within their radius, nor are sufficiently close to any *core point*.



Clustering-based Outlier Detection: Techniques (cont.)

- DBSCAN [Ester et al., 1996] (cont.)
 - *noise points* are removed for the formation of clusters
 - all *core points* that are within a certain distance of each other are allocated to the same cluster
 - each *border point* is allocated to the cluster of the nearest *core points*
 - *noise points* are identified as outliers.



Clustering-based Outlier Detection: Techniques (cont.)

- FindCBLOF [He et al., 2003]
 - To each point, assign a *cluster-based local outlier factor* (CBLOF)
 - The CBLOF score of a point p is determined by the size of the cluster to which p belongs, and the distance between p and
 - its cluster centroid, if p belongs to a large cluster
 - its closest large cluster centroid, if p belongs to a small cluster.
- OR_H [Torgo, 2007]
 - Obtain an agglomerative hierarchical clustering of the data set
 - Use the information on the “path” of each point through the dendrogram as a form to determine its degree of outlyingness

Disadvantages

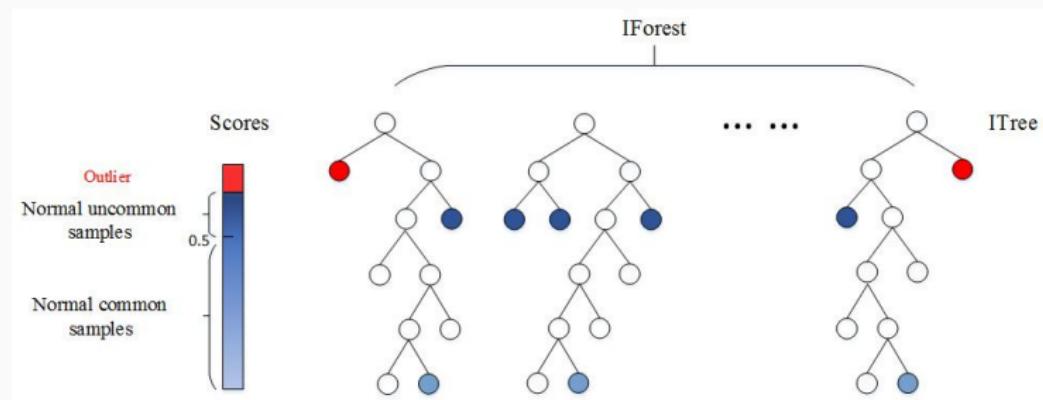
- Computationally expensive in the training phase.
- If normal points do not create any clusters, this technique may fail.
- In high dimensional spaces, clustering algorithms may not give any meaningful clusters.
- Some techniques detect outliers as a byproduct, i.e. they are not optimized to find outliers, their main aim is to find clusters.

Isolation Forest

- iForest [Liu et al., 2008] detects outliers purely based on the concept of isolation without employing any distance or density measure.
- Isolation: separating an instance from the rest of the instances
- A two-stage process.
 1. The first (training) stage builds an ensemble of data-induced random binary decision trees (isolation trees) using sub-samples of the given training set.
 2. The second (evaluation) stage passes test instances through isolation trees to obtain an outlier score for each instance.
- Parameters: number of trees and subsampling size

Isolation Forest (cont.)

- The score is related to average path length
 - outliers are more likely to be isolated closer to the root
 - normal points are more likely to be isolated at the deeper levels



Source: <https://github.com/zmzhang/IOS/blob/master/images/IOS.jpg>

Isolation Forest (cont.)

Advantages

- No distance or density measures to detect anomalies;
- Eliminates a major computational cost of distance calculation in all distance-based and density-based methods;
- Scales up to handle extremely large data size and high-dimensional problems with a large number of irrelevant attributes.

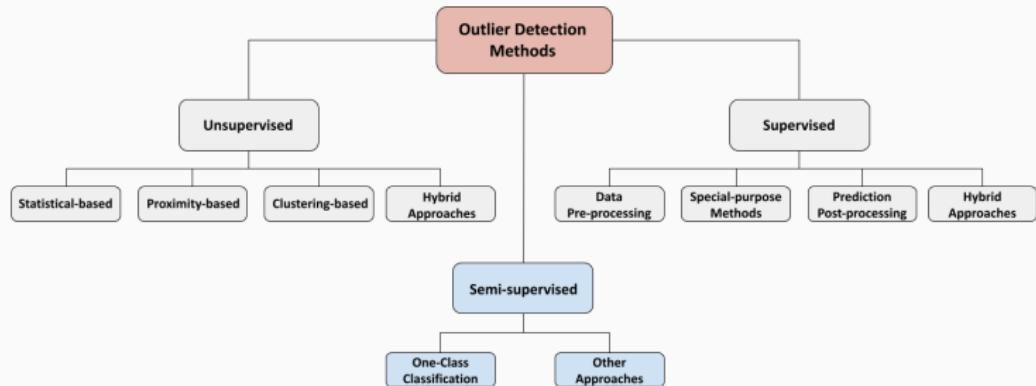
Disadvantages

- Hyperparameters that must be tuned;
- Randomness component: different runs can give different results;
- Large sample sizes may cause masking or swamping.

Outlier Detection Approaches

Semi-supervised Learning Techniques

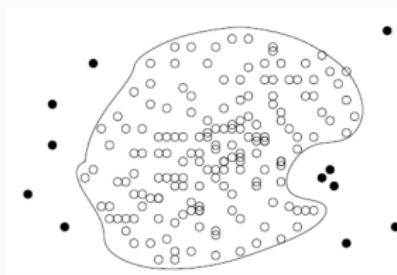
Taxonomy of Outlier Detection Methods



One Class Classification

Proposal

- Build a prediction model to the normal behaviour and classify any deviations from this behaviour as outliers.

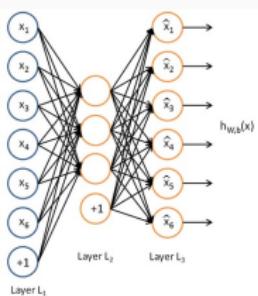


Advantages

- Models are interpretable.
- Normal behaviour can be accurately learned.
- Can detect new outliers that may not appear close to any outlier points in the training set.

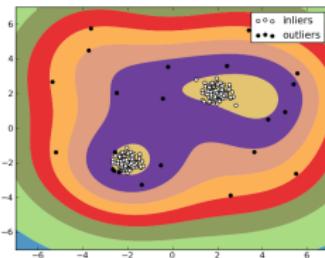
One Class Classification: Techniques

- Auto-associative neural networks [Japkowicz et al., 1995]
 - A feed-forward perceptron-based network is trained with normal data only.
 - The network has the same number of input and output nodes and a decreased number of hidden nodes to induce a bottleneck.
 - This bottleneck reduces the redundancies and focus on the key attributes of data.
 - After training, the output nodes recreate the example given as input nodes.
 - The network will successfully recreate normal data but will generate a high-recreation error for outlier data.



One Class Classification: Techniques (cont.)

- One-class SVM [Tax and Duin, 2004]
 - It obtains a spherical boundary, in the feature space, around the normal data. The volume of this hypersphere is minimized, to minimize the effect of incorporating outliers in the solution.
 - The resulting hypersphere is characterized by a centre \mathbf{c} and a radius R .
 - The optimization problem consists of minimizing the volume of the hypersphere, so that includes all the training points.
 - Every point lying outside this hypersphere is an outlier.



Disadvantages

- Requires previous labeled instances for normal behaviour.
- Possible high false alarm rate - previously unseen normal data may be identified as an outlier.

Outlier Detection Approaches

Advanced Topics

Proposal

- If a data instance is an outlier in a specific context (but not otherwise), then it is considered as a contextual outlier.
- Each data instance is defined using two sets of attributes:
 - **Contextual attributes** used to determine the context (or neighbourhood) for that instance.
 - Sequential Context: position, time.
 - Spatial Context: latitude, longitude.
 - Graph Context: weights, edges.
 - **Behavioural attributes** which define the non-contextual characteristics of an instance.
- The outlier behaviour is determined using the values for the behavioural attributes within a specific context.

Contextual Outlier Detection (cont.)

Example:

- Detect outlier customers in the context of customer groups
 - Contextual attributes: age group, postal code
 - Behavioural attributes: the number of transactions per year, annual total transaction amount

Advantages

- Allow a natural definition of outlier in many real-life applications.
- Detects outliers that are hard to detect when analyzed in the global perspective.

Techniques

- Reduction to point outlier detection
 - Segment data using contextual attributes.
 - Apply a traditional point outlier within each context using behavioural attributes.
 - Model “normal” behaviour with respect to contexts: an object is an outlier if its behavioural attributes significantly deviate from the values predicted by the model.
- Utilizing structure in data
 - Build models from the data using contextual attributes to predict the expected behaviour with respect to a given context.
 - Avoids explicit identification of specific contexts

Disadvantages

- Identifying a set of good contextual attributes.
- It assumes that all normal instances within a context will be similar (in terms of behavioural attributes), while the outliers will be different.

Proposal

- If a collection of related data instances is anomalous with respect to the entire data set, then it is considered a collective outlier.
- The individual data instances in a collective outlier may not be outliers by themselves, but their occurrence together as a collection is anomalous.

Advantages

- Allow a natural definition of outlier in many real-life applications in which data instances are related.

Techniques

- A collective outlier can also be a contextual outlier if analyzed with respect to a context.
- A collective outlier detection problem can be transformed to a contextual outlier detection problem by incorporating the context information.

Disadvantages

- Contrary to contextual outliers, the structures are often not explicitly defined, and have to be discovered as part of the outlier detection process.
- Need to extract features by examining the structure of the dataset, i.e. the relationship among data instances for:
 - sequence data to detect anomalous sequences;
 - spatial data to detect anomalous sub-regions;
 - graph data to detect anomalous sub-graphs.
- The exploration of structures in data typically uses heuristics, and thus may be application dependent.
- The computational cost is often high due to the sophisticated mining process.

Outlier Detection in High Dimensional Data

Challenges

- Interpretation of outliers
 - Detecting outliers without saying why they are outliers is not very useful in high-D due to the many features (or dimensions) involved
 - Identify the subspaces that manifest the outliers
- Data sparsity
 - Data in high-D spaces is often sparse
 - The distance between objects becomes heavily dominated by noise as the dimensionality increases
- Data subspaces
 - Capturing the local behavior of data
- Scalable with respect to dimensionality
 - # of subspaces increases exponentially

Techniques

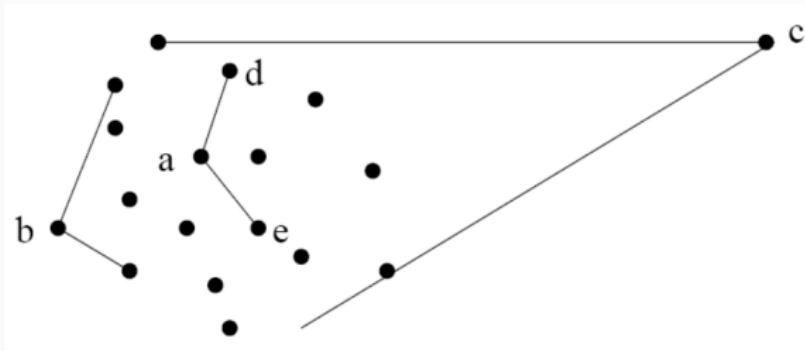
- Find distance-based outliers, but use the ranks of distance instead of the absolute distance in outlier detection.
- Dimensionality reduction: the principal components with low variance are preferred because, on such dimensions, normal objects are likely close to each other and outliers often deviate from the majority.
- Project data onto various subspaces to find an area whose density is much lower than average.

Outlier Detection in High Dimensional Data (cont.)

Techniques (cont.)

- Develop new models for high-dimensional outliers directly. Avoid proximity measures and adopt new heuristics that do not deteriorate in high-dimensional data.

E.g. Angle-based outliers.



Summary

Summary

- Outliers are not necessarily random noise. They can represent critical information that can trigger preventive or corrective actions.
- The interpretability of an outlier detection method is extremely important.
- The nature of the outlier detection problem is dependent on the application domain.
- Different approaches to this problem are necessary.
- Contextual and collective outliers are having increasing applicability in several real-world domains.
- Online Outlier Detection and Distributed Outlier Detection are emerging topics.
- There is much space for the development of new techniques in this area.

References

References

-  Aggarwal, C. (2013).
Outlier Analysis.
Springer New York.
-  Aggarwal, C. C. (2015).
Data Mining, The Texbook.
Springer.
-  Aminian, E., Ribeiro, R. P., and Gama, J. (2021).
Chebyshev approaches for imbalanced data streams regression models.
Data Min. Knowl. Discov., 35(6):2389–2466.
-  Branco, P., Torgo, L., and Ribeiro, R. P. (2016).
A survey of predictive modeling on imbalanced domains.
ACM Comput. Surv., 49(2):31:1–31:50.
-  Branco, P., Torgo, L., and Ribeiro, R. P. (2018).
Resampling with neighbourhood bias on imbalanced domains.
Expert Syst. J. Knowl. Eng., 35(4).
-  Branco, P., Torgo, L., and Ribeiro, R. P. (2019).
Pre-processing approaches for imbalanced distributions in regression.
Neurocomputing, 343:76–99.

References (cont.)

-  Breunig, M. M., Kriegel, H. P., Ng, R., and Sander, J. (2000).
Lof: Identifying density-based local outliers.
In *Proceedings of ACM SIGMOD 2000 International Conference on Management of Data*. ACM Press.
-  Chandola, V., Banerjee, A., and Kumar, V. (2009).
Anomaly detection: A survey.
ACM Computing Surveys (CSUR), 41(3):15.
-  Chawla, N. V., Bowyer, K. W., Hall, O. L., , and Kegelmeyer, W. P. (2002).
Smote: Synthetic minority over-sampling technique.
Journal of Artificial Intelligence Research, 16:321–357.
AAAI Press.
-  Davari, N., Veloso, B., de Assis Costa, G., Pereira, P. M., Ribeiro, R. P., and Gama, J. (2021a).
A survey on data-driven predictive maintenance for the railway industry.
Sensors, 21(17):5739.
-  Davari, N., Veloso, B., Ribeiro, R. P., Pereira, P. M., and Gama, J. (2021b).
Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry.
In *8th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2021, Porto, Portugal, October 6-9, 2021*, pages 1–10. IEEE.

References (cont.)

-  Ester, M., peter Kriegel, H., S, J., and Xu, X. (1996).
A density-based algorithm for discovering clusters in large spatial databases with noise.
pages 226–231. AAAI Press.
-  Han, J., Kamber, M., and Pei, J. (2011).
Data Mining: Concepts and Techniques.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.
-  Hawkins, D. M. (1980).
Identification of Outliers.
Chapman and Hall.
-  He, Z., Xu, X., and Deng, S. (2003).
Discovering cluster based local outliers.
Pattern Recognition Letters, 2003:9–10.
-  Hempstalk, K., Frank, E., and Witten, I. H. (2008).
One-class classification by combining density and class probability estimation.
In *ECML/PKDD (1)*, pages 505–519.
-  Hodge, V. J. and Austin, J. (2004).
A survey of outlier detection methodologies.
Artificial Intelligence Review, 22:2004.

References (cont.)

-  Japkowicz, N., Myers, C., and Gluck, M. A. (1995).
A novelty detection approach to classification.
In *IJCAI*, pages 518–523. Morgan Kaufmann.
-  Joshi, M. V., Agarwal, R. C., and Kumar, V. (2002).
Predicting rare classes: Comparing two-phase rule induction to cost-sensitive boosting.
In *PKDD'02: Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 2431 of *LNCS*, pages 237–249. Springer.
-  Joshi, M. V., Kumar, V., and Agarwal, R. C. (2001).
Evaluating boosting algorithms to classify rare classes: Comparison and improvements.
In *Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2 December 2001, San Jose, California, USA*, pages 257–264.
-  Knorr, E. M. and Ng, R. T. (1998).
Algorithms for mining distance-based outliers in large datasets.
In *VLDB'98: Proceedings of 24th International Conference on Very Large Data Bases*, pages 392–403. Morgan Kaufmann, San Francisco, CA.

References (cont.)

-  Kubat, M. and Matwin, S. (1997).
Addressing the curse of imbalanced training sets: one-sided selection.
In *Proc. 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann.
-  Lazarevic, A. (2008).
Anomaly detection: A tutorial.
Tutorial Session on 2008 Siam Conference on Data Mining (SDM08).
-  Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008).
Isolation forest.
In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422.
-  Maloof, M. A. (2003).
Learning when data sets are imbalanced and when costs are unequal and unknown.
In *ICML-2003 workshop on learning from imbalanced data sets II*, volume 2, pages 2–1.
-  Moniz, N., Ribeiro, R. P., Cerqueira, V., and Chawla, N. V. (2018).
Smoteboost for regression: Improving the prediction of extreme values.
In *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018*, pages 150–159. IEEE.

References (cont.)

-  Portela, E., Ribeiro, R. P., and Gama, J. (2017).
Outliers and the simpson's paradox.
In *Advances in Soft Computing - 16th Mexican International Conference on Artificial Intelligence, MICAI 2017*, volume 10632 of *LNCS*, pages 267–278. Springer.
-  Portela, E., Ribeiro, R. P., and Gama, J. (2019).
The search of conditional outliers.
Intell. Data Anal., 23(1):23–39.
-  Ribeiro, R. P. (2011).
Utility-based Regression.
PhD thesis, Dep. Computer Science, Faculty of Sciences - University of Porto.
-  Ribeiro, R. P. and Moniz, N. (2020).
Imbalanced regression and extreme value prediction.
Mach. Learn., 109(9-10):1803–1835.
-  Ribeiro, R. P., Pereira, P. M., and Gama, J. (2016).
Sequential anomalies: a study in the railway industry.
Mach. Learn., 105(1):127–153.
-  Tax, D. (2001).
One-class classification: Concept learning in the absence of counter-examples.
PhD thesis, Technische Universiteit Delft.

References (cont.)

-  Tax, D. M. J. and Duin, R. P. W. (2004).
Support vector data description.
Machine Learning, 54(1):45–66.
-  Torgo, L. (2007).
Resource-bounded fraud detection.
In *Progress in Artificial Intelligence, 13th Portuguese Conference on Artificial Intelligence, EPIA 2007, Workshops*, pages 449–460.
-  Torgo, L. (2016).
Outlier detection methods.
Slides.
-  Torgo, L., Ribeiro, R. P., Pfahringer, B., and Branco, P. (2013).
Smote for regression.
In *Progress in Artificial Intelligence*, pages 378–389. Springer.
-  Weiss, G. M. (2004).
Mining with rarity: a unifying framework.
SIGKDD Explorations Newsletter, 6(1):7–19.
-  Zhang, Y., Meratnia, N., and Havinga, P. (2007).
A taxonomy framework for unsupervised outlier detection techniques for multi-type data sets.

ANN + Introduction to Deep Learning

Rita P. Ribeiro

Machine Learning - 2021/2022



DEPARTAMENTO DE CIÉNCIA DE COMPUTADORES
FACULDADE DE CIÉNCIAS DA UNIVERSIDADE DO PORTO

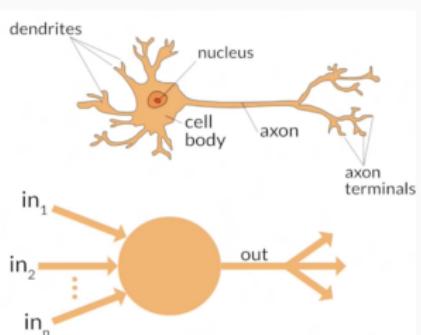
Summary

- Artificial Neural Networks
- (Very Short) Introduction to Deep Learning

Artificial Neural Networks

Artificial Neural Networks (ANN)

- Models with a strong biological inspiration. The brain is a highly complex structure, non linear and highly parallel.
- McCulloch e Pitts (1943) proposed the first artificial model of a neuron.
- Neuron: many-inputs / one-output unit
- Synapses: electrochemical contact between neurons

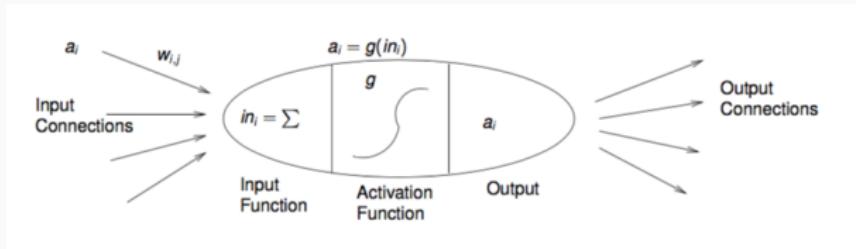


- Output of a neuron: excited or not excited
- Incoming signals from other neurons determine if the neuron shall excite ("fire")
- Output subject to attenuation in the synapses

Artificial Neural Networks (ANN) (cont.)

- An artificial neural network is composed by a set of units (neurons) that are connected. These connections have an associated weight.
- Each unit has an activation level as well as means to update this level.
- Some units are connected to the outside world. We have input and output neurons.
- Learning within ANNs consists of updating the weights of the network connections.

Artificial Neural Networks: Artificial Neuron



- Each unit has a very simple function:
 - receive the input impulses and calculate its output as a function of these impulses.
- This calculation is divided in two parts:
 - a linear combination of the inputs: $in_i = \sum_j w_{ij} a_j + b$
 - a (typically) non-linear activation function: $a_i = g(in_i)$

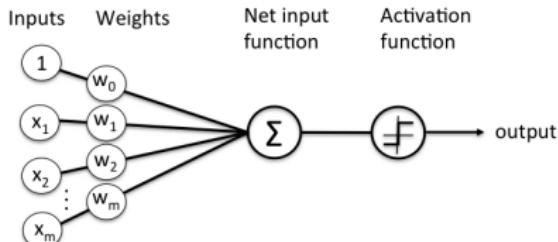
Artificial Neural Networks: Perceptron

- Rosenblatt (1958) introduced the notion of perceptron networks. This work was then further extended by Minsky and Papert (1969).
- **Perceptrons** are networks with an input layer and an output layer.



Artificial Neural Networks: Perceptron (cont.)

Simplest Perceptron



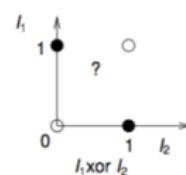
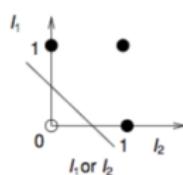
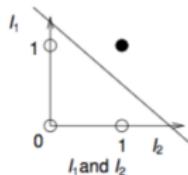
Schematic of Rosenblatt's perceptron.

- A linear classifier for binary classification problems

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + w_0 > 0 \\ 0 & \text{otherwise} \end{cases}$$

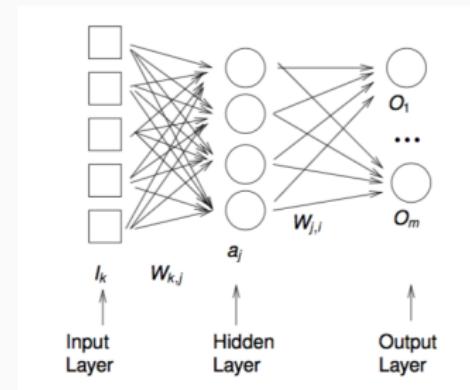
- It learns by updating the weights through delta rule with learning rate η
- $w_i(t+1) = w_i(t) + \eta(\text{true} - \text{predicted})x_i$

Perceptrons are limited to linearly separable functions.



Artificial Neural Networks: Types of ANNs

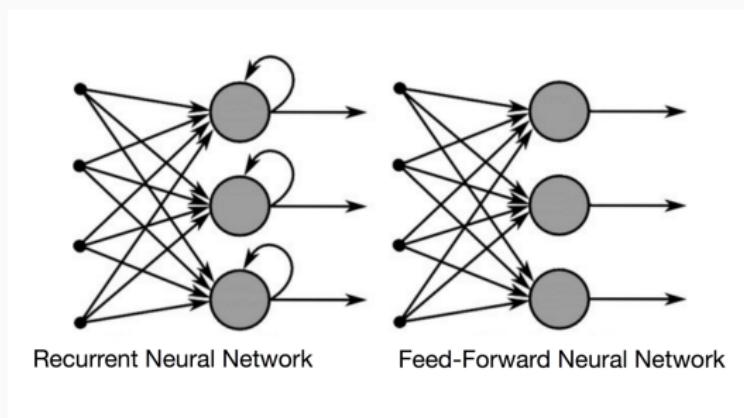
- Feed-forward networks (Multilayer perceptrons)
 - networks with uni-directional connections (from input to output), and without cycles
 - each unit is connected only to units in the following layer
 - there are not connections from units on a certain layer and units on previous layers



Artificial Neural Networks: Types of ANNs (cont.)

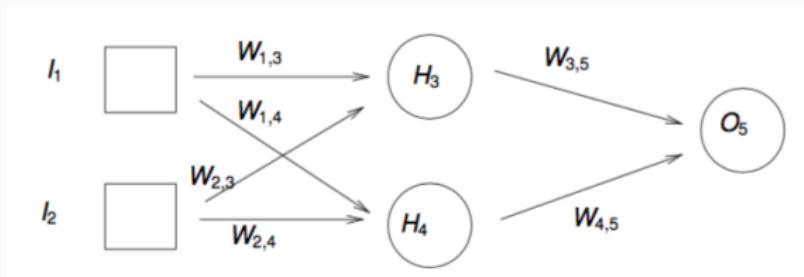
- Recurrent networks

- networks with arbitrary connections
- due to the possible feedback effects, recurrent networks are potentially more unstable, possibly exhibiting chaotic behaviors
- usually they take longer to converge



Artificial Neural Networks: Types of ANNs (cont.)

- Example of a feed-forward network with one input layer (I), one hidden layer (H) and one output layer (O) with one output variable.



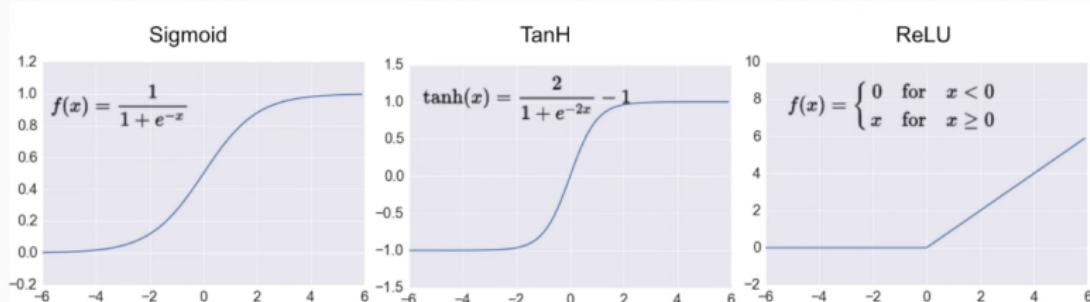
- The output can be represented as follows:

$$\begin{aligned}a_5 &= g(W_{3,5}a_3 + W_{4,5}a_4) = \\&= g(W_{3,5}g(W_{1,3}a_1 + W_{2,3}a_2) + W_{4,5}g(W_{1,4}a_1 + W_{2,4}a_2))\end{aligned}$$

- where $g()$ is the activation function

Artificial Neural Networks: Activation Functions

- Activation functions are used to determine the output of each node of the neural network
 - linear
 - non-linear: most commonly used as it allows the model to generalize or adapt with variety of data
- Examples



Artificial Neural Networks: Backpropagation Algorithm

- This is the most popular algorithm for learning ANNs.
- It has similarities with the learning algorithm used in perceptron networks
- **Intuition:**
 - each unit is responsible for a certain fraction of the error in the output nodes to which it is connected
 - thus, the error is divided according to the weight of the connection between the respective hidden and output units, thus propagating the errors backwards
- Backpropagation computes the gradient in weight space of a feedforward neural network, with respect to a loss function.

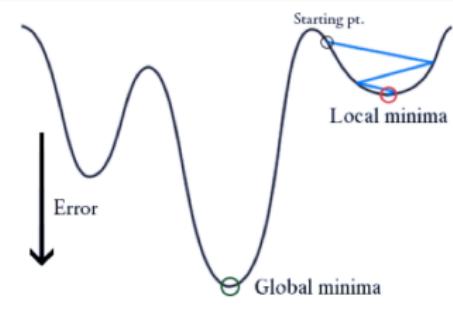
Artificial Neural Networks: Backpropagation Algorithm (cont.)

The Algorithm (for one hidden layer)

- Initialize network weights (often small random values)
- Do
 - For each example in training set
 - predict the output
 - calculate the prediction error by a loss function
 - compute δ_h for all the weights from hidden layer to output layer
 - compute δ_i for all the weights from input layer to hidden layer
 - update network weights
 - Until it converges
 - all examples are classified correctly or stopping criterion is satisfied
 - Return the network

Artificial Neural Networks: Backpropagation Algorithm (cont.)

Gradient Descendent



- **Stochastic Gradient Descent:** instead of calculating the gradient of the full error function (which involves using the full training set), we update the weights one example at a time.
- **Batch Gradient Descent:** the batch size is the number of sub samples given to the network after which weights update happens.
- Both are more effective to escape from local minima.

When to stop training?

- If stopping too early: risk of getting a network not yet trained.
- If stopping too late: danger of overfitting (adjustment to noise in the data)
- Stopping criteria:
 - maximum number of iterations
 - error based on the training set
 - when the error in the training set is below a certain limit.
 - error based on a validation set (independent from the training set)
 - when the error on the validation set has reached a minimum.

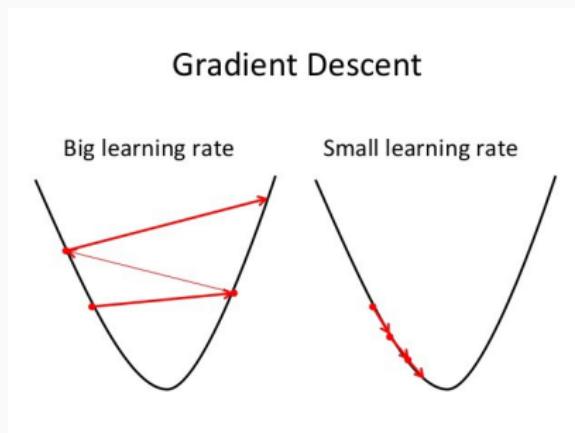
Artificial Neural Networks: Issues

Network topology

- The number of nodes in the hidden layer
 - few nodes: underfitting
 - many nodes: overfitting
 - there are no criteria for defining the number of nodes in the hidden layer
- Effect of learning rate
 - a small learning rate has the effect of learning times higher
 - a high learning rate may lead to non-convergence

Artificial Neural Networks: Issues (cont.)

- The **learning rate** sets the size of the steps to obtain the direction of maximum descendent.



Generalization vs Specialization trade-off

- Optimal number of hidden neurons
 - too many hidden neurons: you get an overfit, training set is memorized, thus making the network useless on new data sets
 - not enough hidden neurons: network is unable to learn problem concept
- Overtraining
 - too much examples, the ANN memorizes the examples instead of the general idea

Some relevant hyperparameters

- Network Structure

- number of layers
- number of neurons in each layer
- weights initialization
- activation function

- Training Algorithm

- learning rate
- number of epochs
- early stopping criterion
- weight decay (a regularization on the network weights)

Some Tips

- Features with very different distributions of values are not convenient, given the typical activation functions.
 - Data should be standarized.
- Missing values in input features may be represented as zeros, which do not influence the neural net training process.
- Output in Multiclass Setting
 - Use one-hot encoding, there are M output neurons (1 per class),
 - For each case, the class with the highest probability value.

Some Tips (cont.)

- Initialize the weights with small random values $[-0.05, 0.05]$
- Shuffle the training set between epochs, i.e. change the sequence of the examples
- The learning rate must start with a high value that decreases progressively
- Train the network several times using different initialization of the weights

Artificial Neural Networks: Wrap-Up

Use ANNs when

- Input is high-dimensional discrete or real-valued (e.g. raw sensor input)
- Output is discrete or real valued
 - Classification: use Softmax function as activation function in output layer to compute the probabilities for the classes
 - Regression: use a linear function as activation function in output layer
- Output is a vector of values
- Possibly noisy data
- Form of target function is unknown
- Human readability of result is unimportant

Artificial Neural Networks: Wrap-Up (cont.)

Pros

- Tolerance of noisy data
- Ability to classify patterns on which they have not been trained
- Successful on a wide range of real-world problems
- Algorithms are inherently parallel

Cons

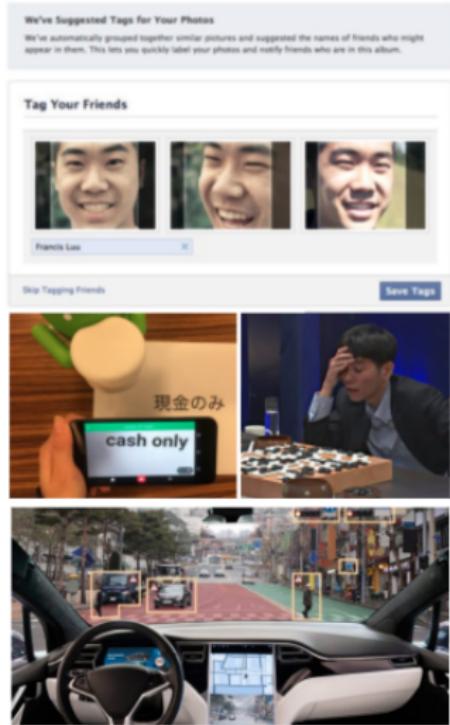
- Long training times
- Resulting models are essentially black boxes

(Very Short) Introduction to Deep Learning

A (Very Short) Introduction to Deep Learning

Deep Learning: where?

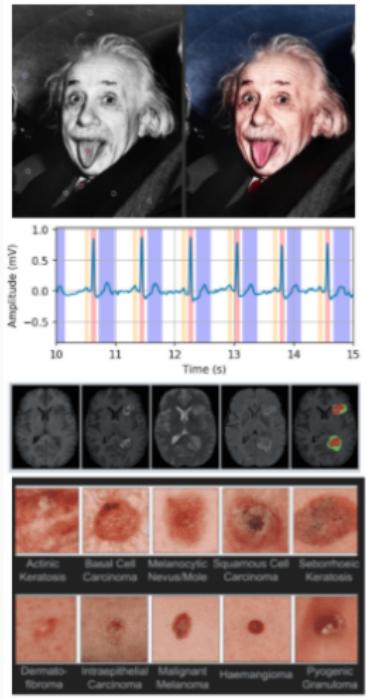
- Image recognition (e.g. Google, Facebook)
- Automatic text translation (e.g. Google Translator)
- Answers in natural language / digital assistants
- Games (e.g. DeepMind AlphaGo)
- Transcript of handwritten text
- Self-driving cars



A (Very Short) Introduction to Deep Learning (cont.)

Deep Learning: where?

- Image colorization, caption generation
- Classification of protein and DNA sequences
- Heart sound: classification and segmentation
- Tumor images detection from MRI, CT, X-rays
- Skin lesion classification from clinical and dermoscopic images
- Parkinson's disease detection from voice recording



A (Very Short) Introduction to Deep Learning (cont.)

- Deep learning = Deep neural networks
 - Deep = high number of hidden layers
 - Learn a larger number of parameters!
- It was made possible recently (~ in the last 6 years) since we have:
 - Access to big amounts of (training) data
 - Increased computational capabilities (e.g., GPUs)

Convolutional neural networks (CNNs)

Convolution Neural Networks (CNN)

- Feedforward neural networks
- Neurons typically use the ReLU or sigmoid activation functions
- Weight multiplications are replaced by convolutions (filters)
- Change of paradigm: can be directly applied to the raw signal, without computing first ad hoc features
- Features are learnt automatically!!

Convolutional neural networks (CNNs) (cont.)

Convolution

- mathematical operation between two matrices;
- the 2nd matrix is a filter that is overlapped to each position of the 1st matrix.

The diagram illustrates a 2D convolution operation. It shows a 6x6 input matrix on the left, a 3x3 filter matrix in the middle, and a 3x3 output matrix on the right. The input matrix has values: [3, 0, 1, 2, 7, 4; 1, 5, 8, 9, 3, 1; 2, 7, 2, 5, 1, 3; 0, 1, 3, 1, 7, 8; 4, 2, 1, 6, 2, 8; 2, 4, 5, 2, 3, 9]. The filter matrix has values: [1, 0, -1; 1, 0, -1; 1, 0, -1]. The output matrix has a single value '5' at position (0,0). A blue arrow points from the formula below to the result '5'. The formula is: $3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times (-1) + 8 \times (-1) + 2 \times (-1) = 5$.

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*
(conv 2D)

1	0	-1
1	0	-1
1	0	-1

=

5		

$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times (-1) + 8 \times (-1) + 2 \times (-1) = 5$

Convolutional neural networks (CNNs) (cont.)

Convolution

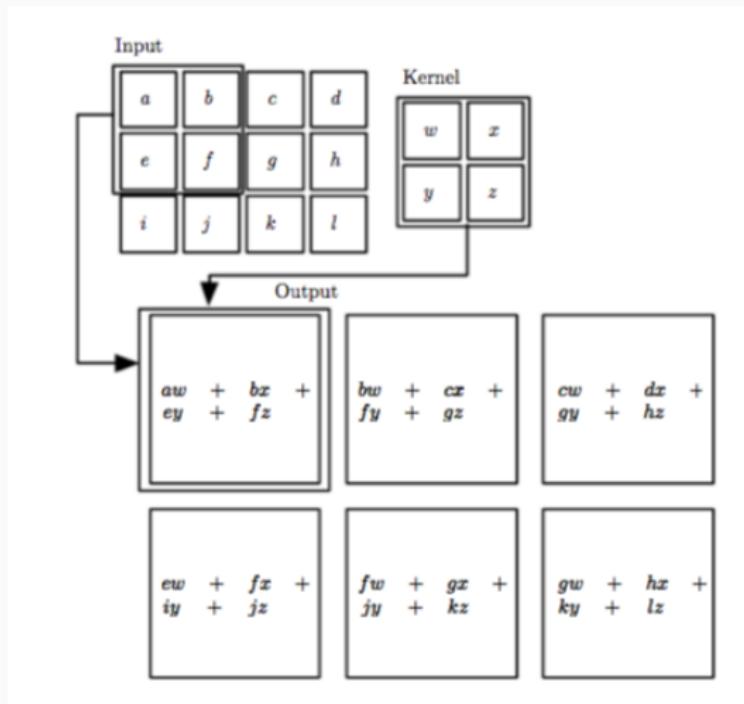
- mathematical operation between two matrices;
- the 2nd matrix is a filter that is overlapped to each position of the 1st matrix.

$$\begin{array}{|c|c|c|c|c|c|} \hline 3 & 0 & 1 & 2 & 7 & 4 \\ \hline 1 & 5 & 8 & 9 & 3 & 1 \\ \hline 2 & 7 & 2 & 5 & 1 & 3 \\ \hline 0 & 1 & 3 & 1 & 7 & 8 \\ \hline 4 & 2 & 1 & 6 & 2 & 8 \\ \hline 2 & 4 & 5 & 2 & 3 & 9 \\ \hline \end{array} \quad * \quad (\text{conv 2D}) \quad = \quad \begin{array}{|c|c|c|c|} \hline 5 & -4 & 0 & 8 \\ \hline -10 & -2 & 2 & 3 \\ \hline 0 & -2 & -4 & -7 \\ \hline -3 & -2 & -3 & -16 \\ \hline \end{array}$$

$0x1 + 5x1 + 7x1 + 1x0 + 8x0 + 2x0 + 2x(-1) + 9x(-1) + 5x(-1) = 5$

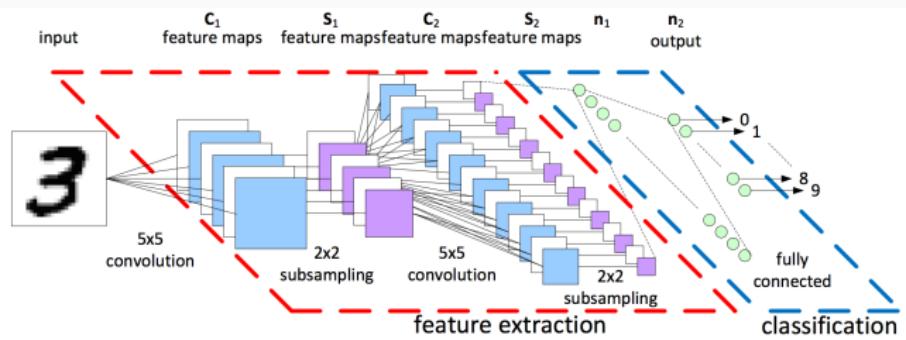
Convolutional neural networks (CNNs) (cont.)

Convolution



Convolutional neural networks (CNNs) (cont.)

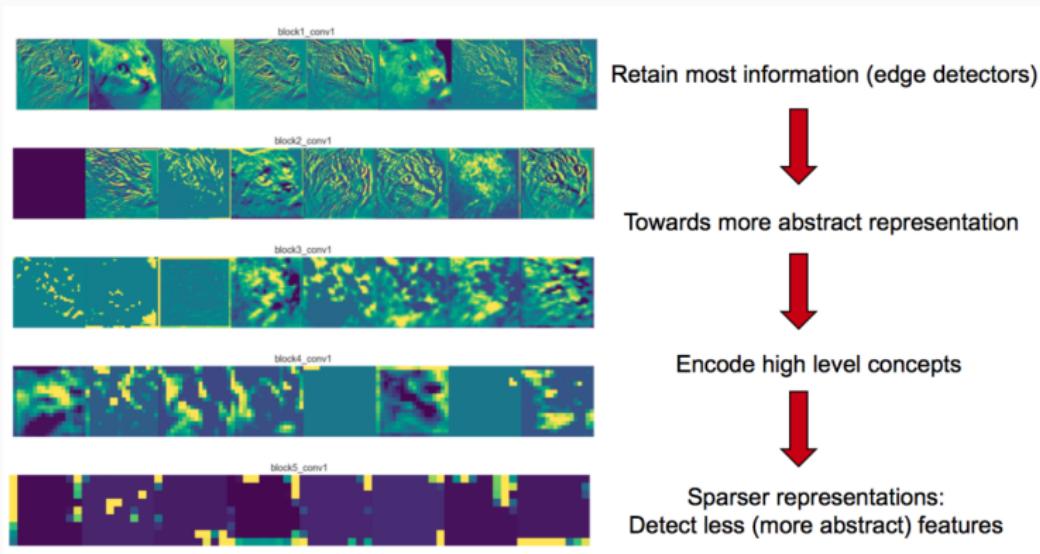
Example for Image Processing



- convolutional layers, followed by nonlinear activation and subsampling (pooling)
- output of hidden layers (feature maps) are features learnt by the CNN
- flatten fully connected layers for classification (as in “standard” NN)

Convolutional neural networks (CNNs) (cont.)

Example for Image Processing: feature extraction

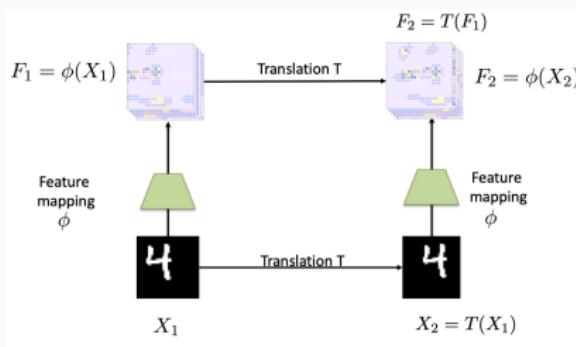


- the convolutions, applied to various zones of the image, act as filters that can detect certain patterns

Convolutional neural networks (CNNs) (cont.)

Properties

- Reduced amount of parameters to learn (local features)
- More efficient than dense multiplication
- Specifically thought for images or data with grid-like topology
- Convolutional layers are equivariant to translation



- if image input is translated by a certain amount,
- the feature map is also translated
- useful for classification

- Currently state-of-the-art in several tasks

(Very Short) Introduction to Deep Learning: Wrap-Up

Great results! But...

- Like any other technique, DL does not solve all problems and will not always be the best option for any learning task.
- Difficult to select best architecture for a problem
- Require new training for each task/configuration
- (Most commonly) require a large training dataset to generalize well
 - Data augmentation, weight regularization, dropout, transfer learning, etc.
- Still not fully understood why it works so well
 - Unstable against adversarial examples

(Very Short) Introduction to Deep Learning: Wrap-Up (cont.)

To know more

- Book – I.Goodfellow,Y.Bengio, and A.Courville. [Deep learning](#). Vol.1. Cambridge: MIT press, 2016.
- Tutorial – Oxford Visual Geometry Group: [VGG Convolutional Neural Networks Practical](#)

References

References

- Aggarwal, Charu C. 2015. *Data Mining, the Texbook*. Springer.
- Gama, João, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. 2015. *Extração de Conhecimento de Dados: Data Mining -3rd Edition*. Edições Sílabo.
- Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Renna, Francesco. 2019. “Introduction to Deep Learning.” Slides.
- Rocha, Miguel. 2019. “Foundations and Applications of Machine Learning Course.” Slides.
- Smola, Alex J., and Bernhard Schölkopf. 2004. “A Tutorial on Support Vector Regression.” *Statistics and Computing* 14 (3): 199–222.
- Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining*. 2nd ed. Pearson.
- Torgo, Luís. 2017. “Data Mining I Course.” Slides.

metalearning and autoML

Machine Learning
(Aprendizagem Computacional)

Carlos Soares

csoares@fe.up.pt

[some slides shamelessly stolen from J. Vanschoren]

plan

- the world where automated ml lives
 - a world of **many models**
 - needs **model management**
 - **metalearning/automl** can help
 - but **opportunities and challenges** are still open

the world where automated ml lives

lots of data

+

lots of detail

+

lots of problems

+

lots of models

=

extreme data mining

(adapted from
Soulié-Fogelman)

+ lots of models

- more specific knowledge
- ... that is, models for smaller subsets
 - e.g. [Fogelman 06]
 - “broadband communications company moved from 5 cross-sell models per year to 1600;
 - A wireless communications company that produces 700 CRM models per year;”
- ... eventually, individual entities
 - e.g. a recommendation model for each customer
 - e.g. soft sensors
 - e.g. UPV’s project with large retail company
 - 50 million models to predict the sales of products

todo

- the world where automated ml lives
 - a world of **many models**
 - needs **model management**
 - **metalearning/automl** can help
 - but **opportunities and challenges** are still open

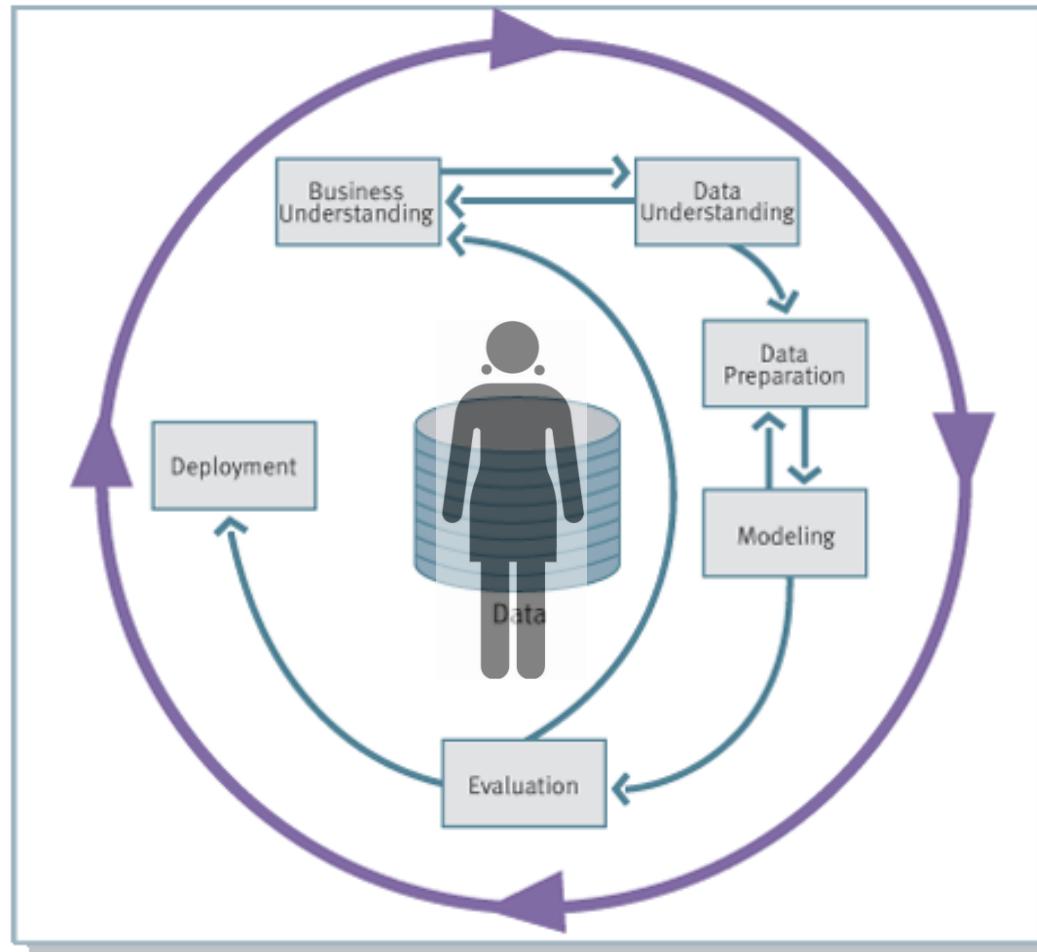
extreme data mining

lots of data
+
lots of detail
+
lots of problems
+
lots of models
=

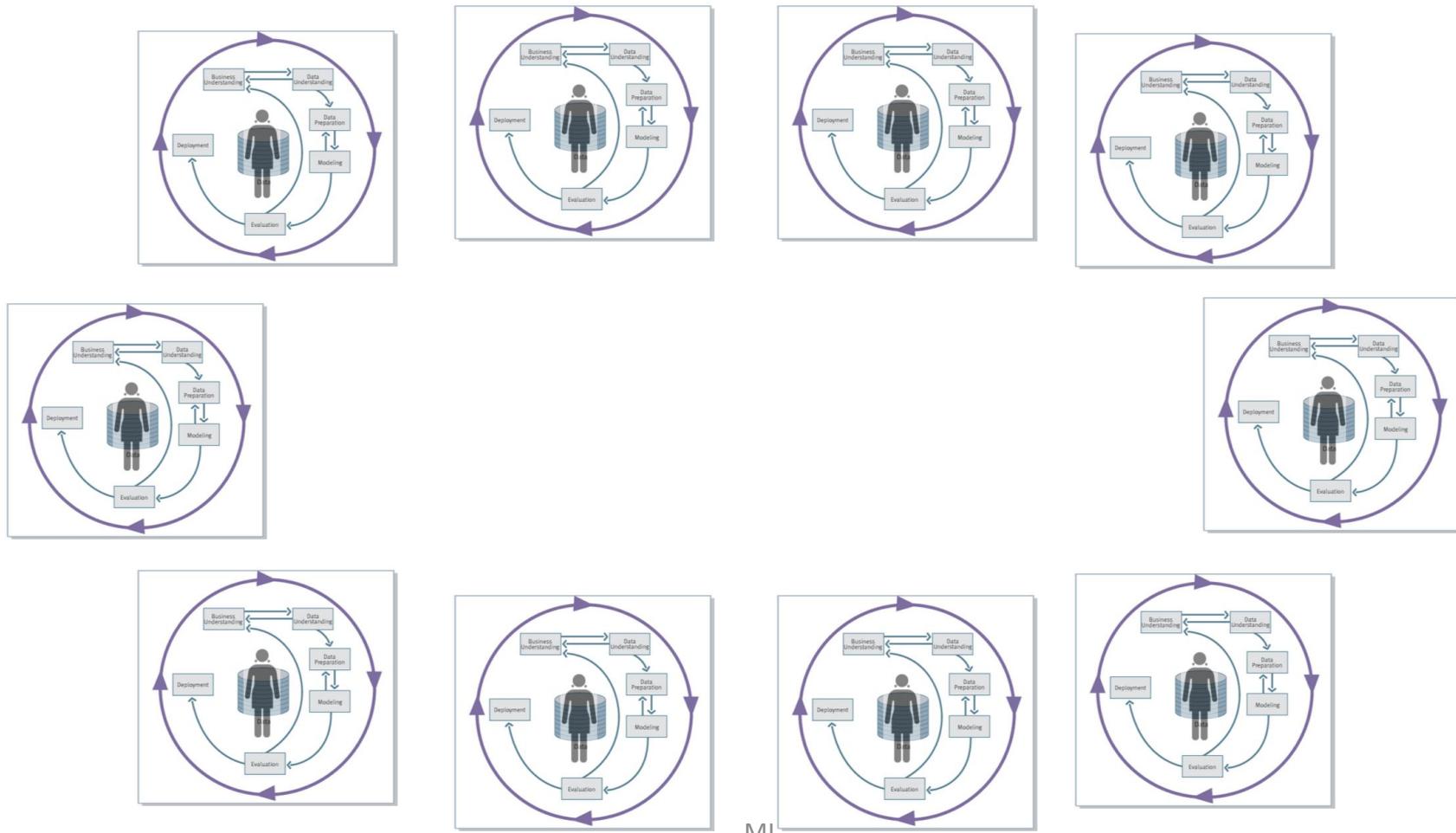
lots of data mining

(adapted from
Soulié-Fogelman)

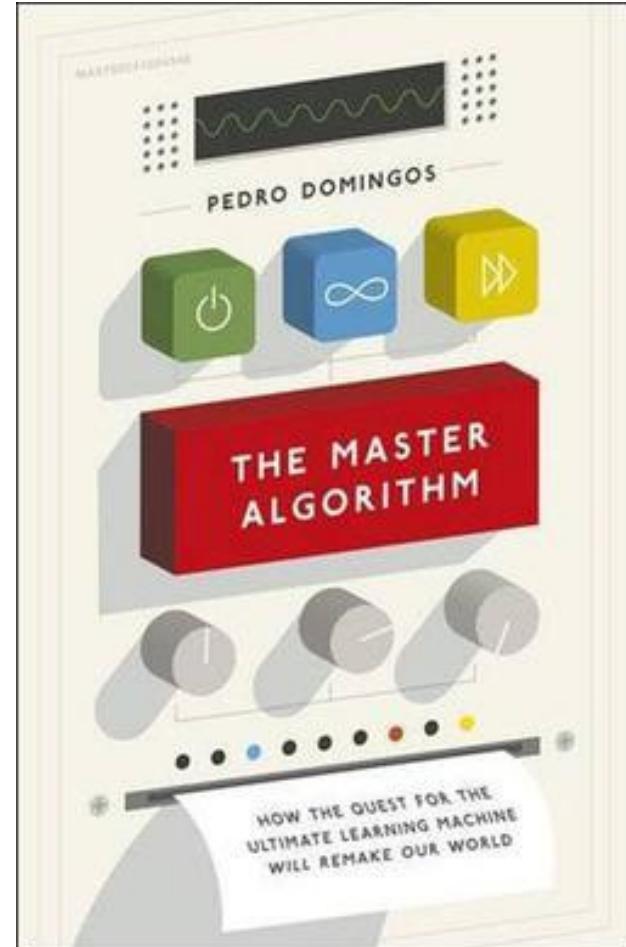
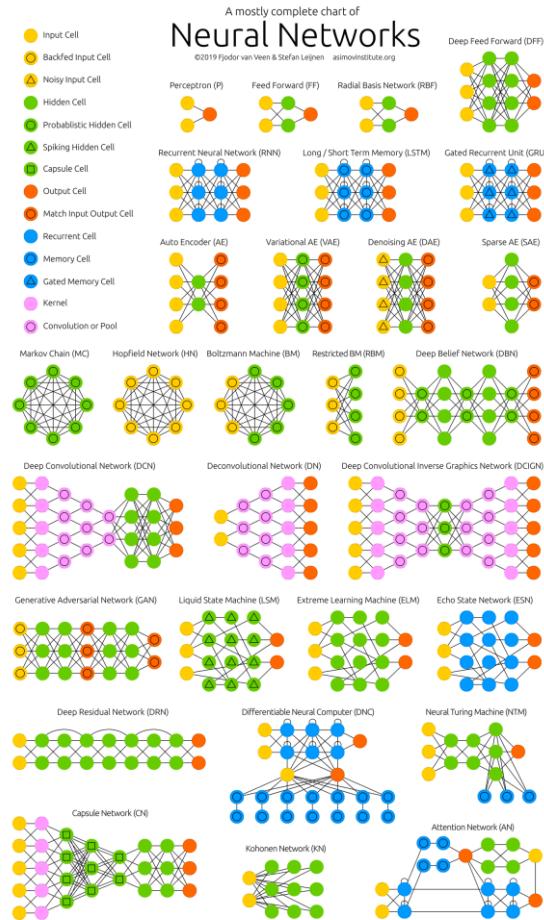
traditional DM methodology



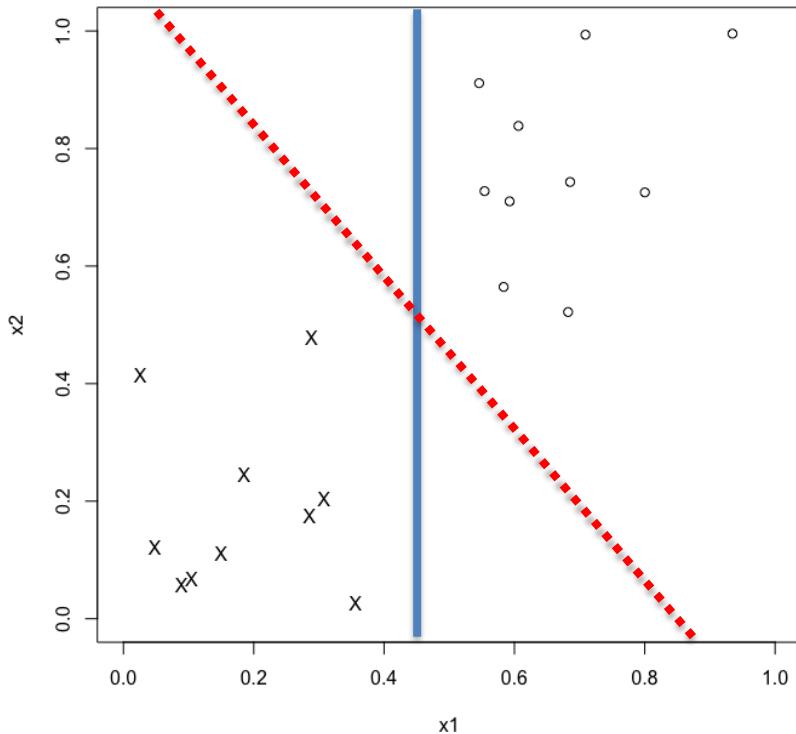
this is not possible!



wait, is this really a problem?



bias: can't live with it



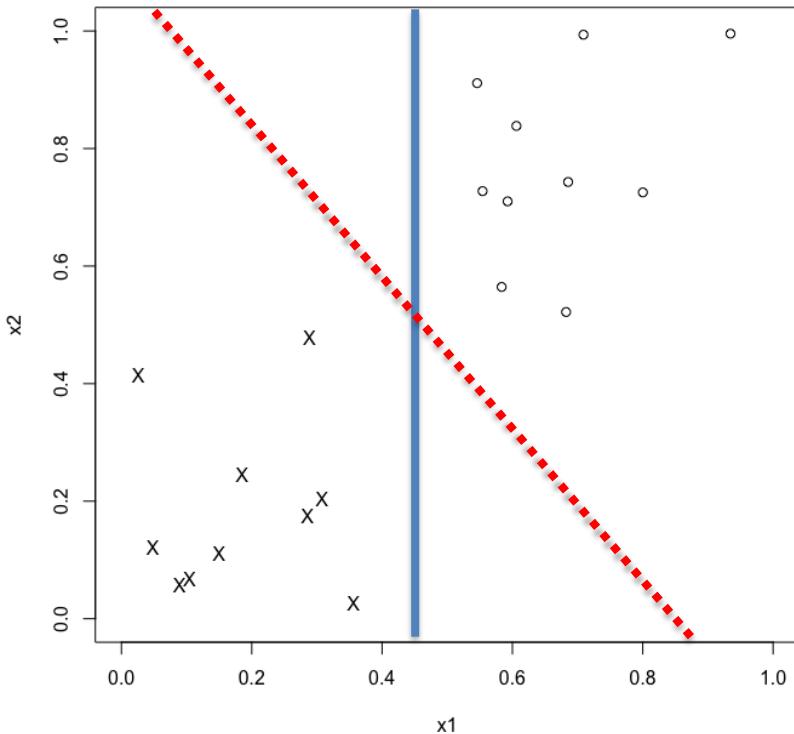
- given
 - dataset
 - learning algorithm
- not every model is possible
 - e.g. DT and LR
 - ... but not DT and LR

... and can't live without it

- bias-free learning is futile
(Mitchell 97, Ch. 2)
 - an algorithm that assumes nothing concerning the function it is trying to learn has no rational basis to classify unknown cases
- bias = criteria to prefer one model relative to another
- ... so, how to select the best model if all models are considered equally suitable?

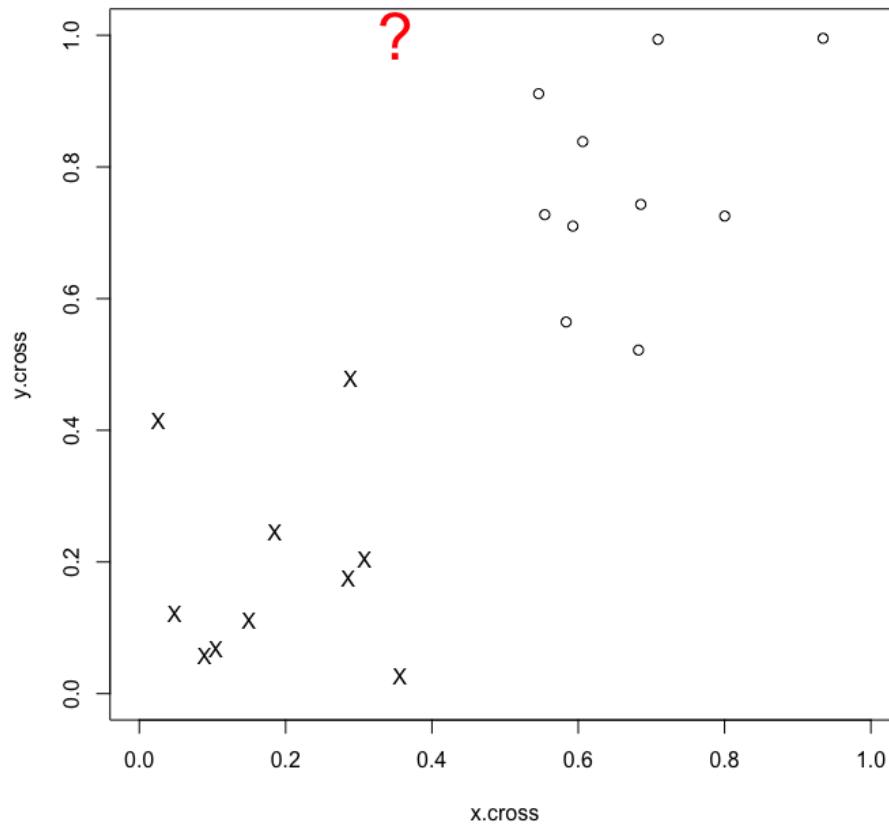


the bias-free algorithm

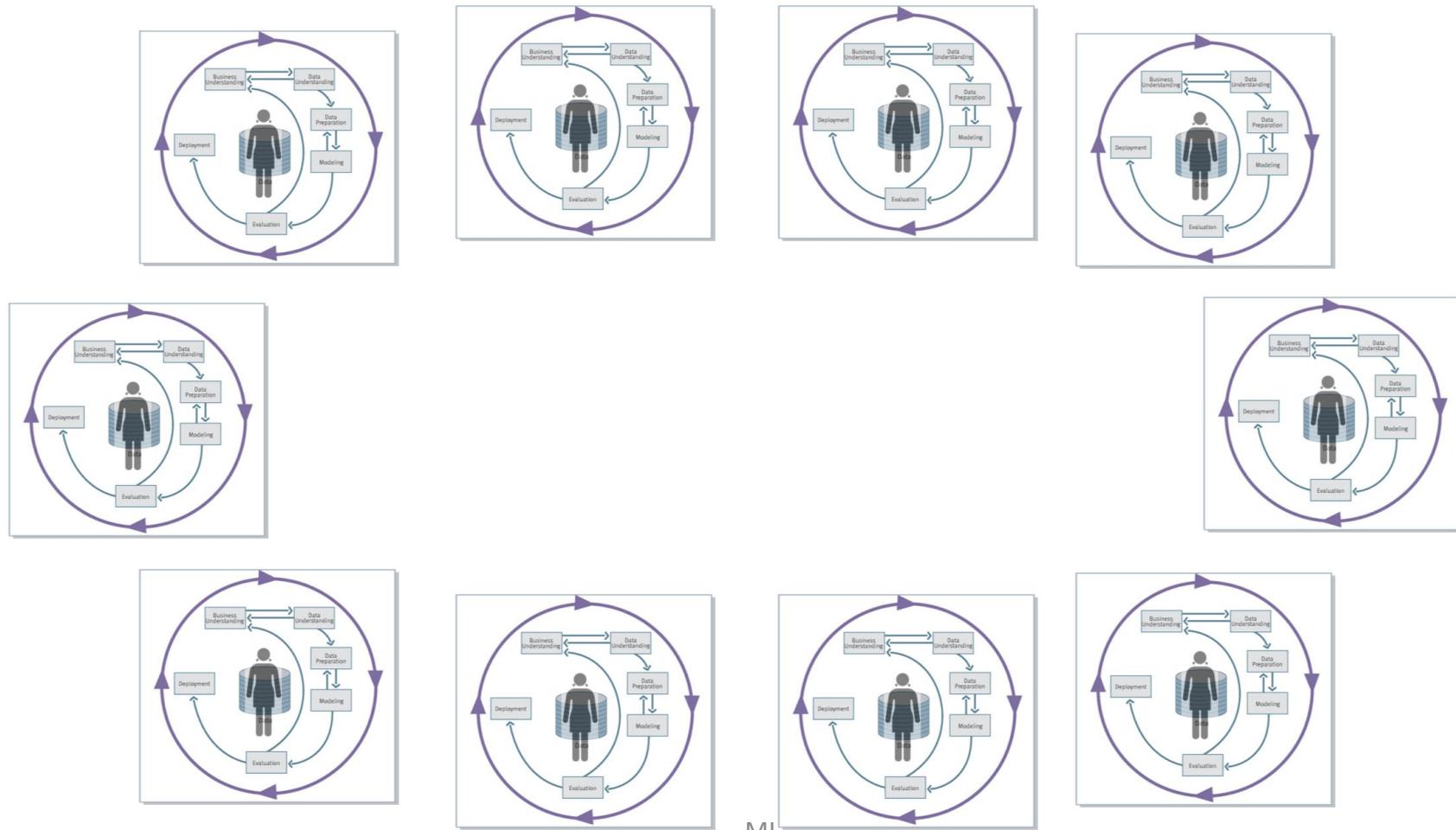


- ... can learn any model
 - e.g. the **DT** and the **LR**
- ... but doesn't have any preference for **one** over the **other**
 - ... or for **one** over the **other**

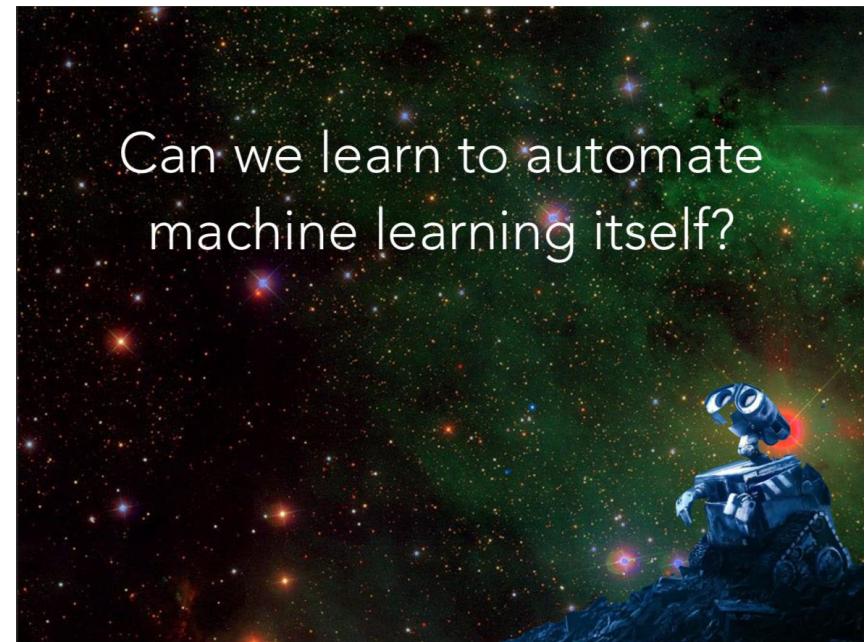
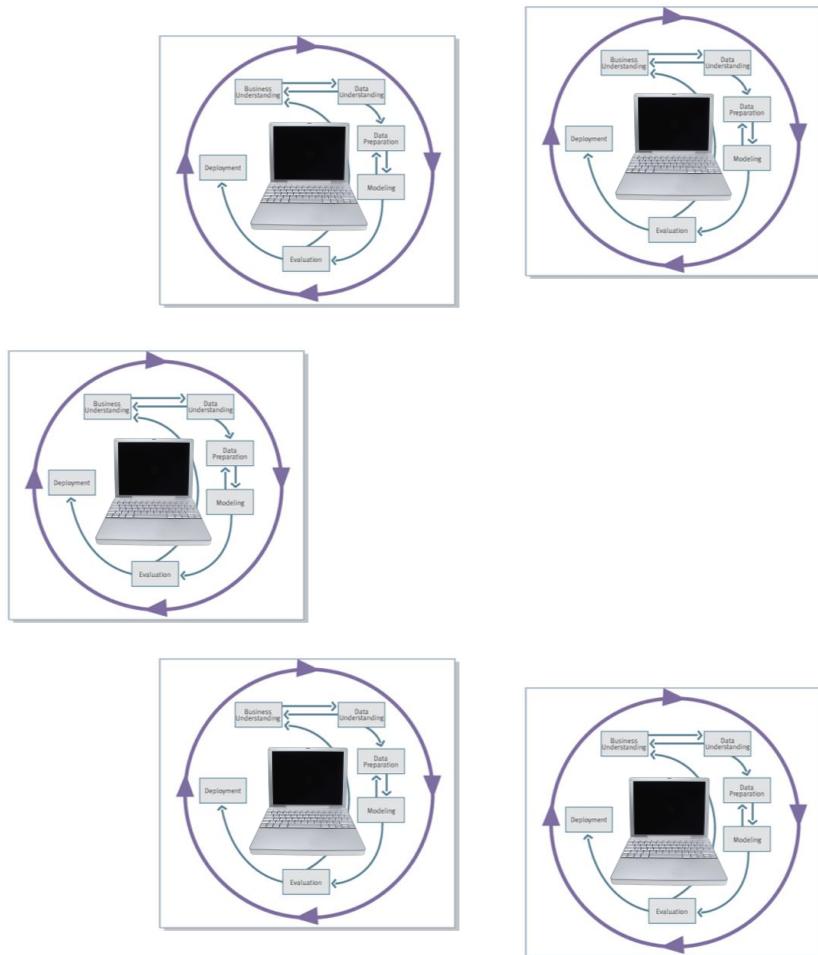
... right?



so, if this is not possible?...



[the dream]



Can we learn to automate
machine learning itself?

shameless plagiarism of someone who prepares more beautiful slides than I do

... but maybe this is...



todo

- the world where automated ml lives
 - a world of **many models**
 - needs **model management**
 - **metalearning/automl** can help
 - but **opportunities and challenges** are still open

a simple autoML problem: algorithm selection



how can I use previous experience to help me
choose the best algorithm?

```
> iris
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1       5.1        3.5       1.4        0.2   setosa
2       4.9        3.0       1.4        0.2   setosa
3       4.7        3.2       1.3        0.2   setosa
4       4.6        3.1       1.5        0.2   setosa
5       5.0        3.6       1.4        0.2   setosa
```

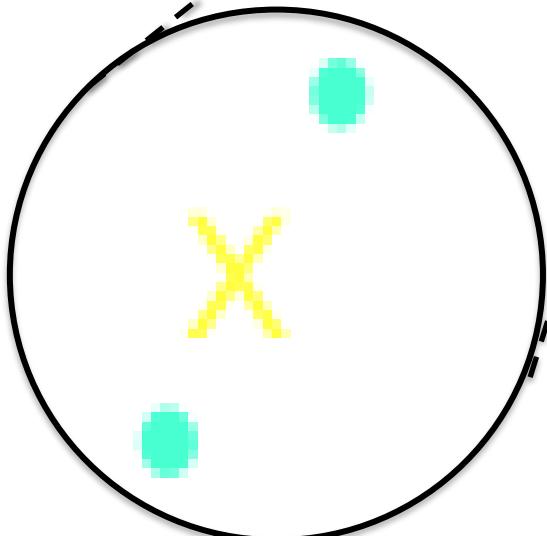
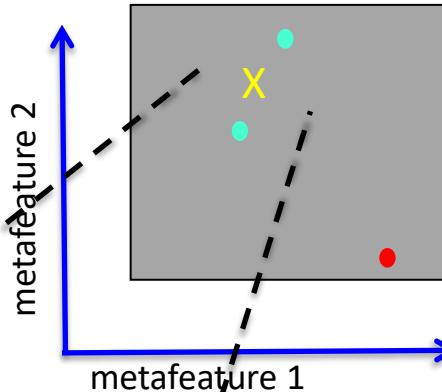
```
> house_votes_84
    V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
1   n  y  n  y  y  y  n  n  n  y <NA>  y  y  y  y  n  y republican
2   n  y  n  y  y  y  n  n  n  n  n  y  y  y  y  y  n <NA> republican
3 <NA> y  y <NA> y  y  n  n  n  y  n  y  y  n  n  democrat
4   n  y  y  n <NA> y  n  n  n  n  y  n  y  n  n  y  democrat
5   v  v  v  v  n  v  n  n  n  n  v <NA>  v  v  v  v  v democrat
```



ML



autoML approaches (1/2): metalearning for algorithm selection



> iris				
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2

n.examples	n.attributes	n.classes	def.accuracy	meta.target
imports_85	205	25	0	0.3268293 DT
ionosphere	351	33	0	0.6410256 ID
iris	150	4	0	0.3333333 DT
KR-vs-KP	3196	36	0	0.5222153 DT
lung-cancer	32	56	0	0.4062500 LR

X =

experimental results: an example

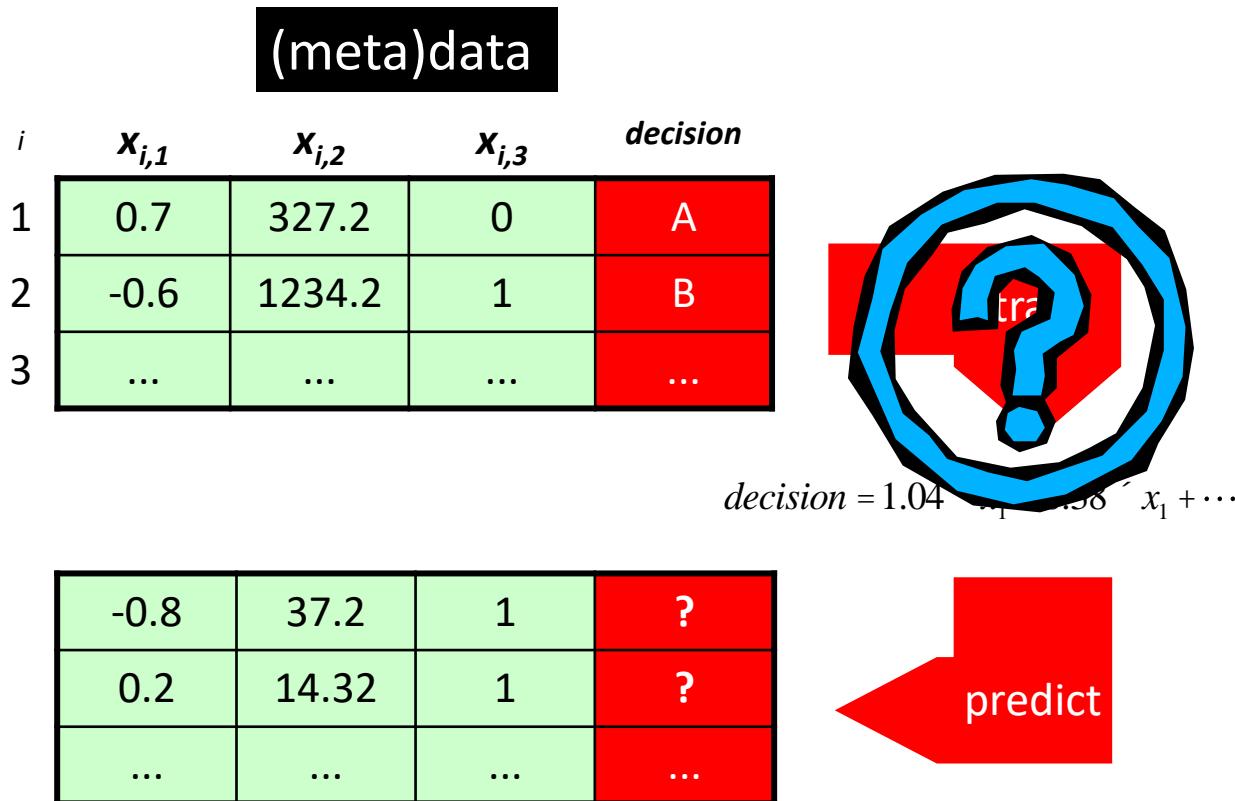
- when to prune decisions trees
 - 3 classes: prune, don't prune, doesn't matter
 - selected **64** datasets from UCI
- metafeatures
 - entropy of classes (target attribute)
 - mean entropy of symbolic attributes
- positive but not excellent
 - very simple example
 - better examples in different contexts

algorithm	accuracy (%)
default	41
dt	41
ld	41
rf	47
svm	41
nn	45

very hard problem

summary

- metalearning for algorithm selection
 - induce model from *metadata* to predict the best algorithm on a new dataset



autoML is old

THE ALGORITHM SELECTION PROBLEM

John R. Rice
Computer Science Department
Purdue University
West Lafayette, Indiana 47907

July 1975

CSD-TR 152

(This is a revised version of CSD-TR 116, 117 and 130)

(To appear in Advances in Computers, Vol. 15, Academic Press, 1976)

autoML is hard

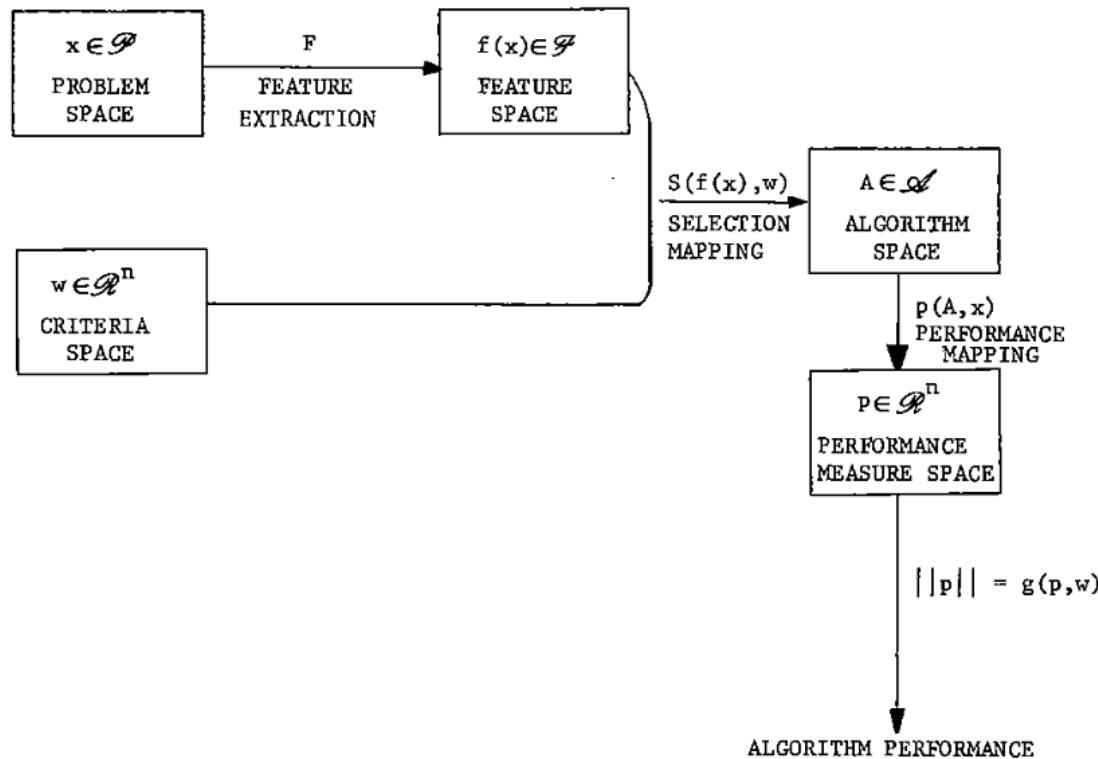
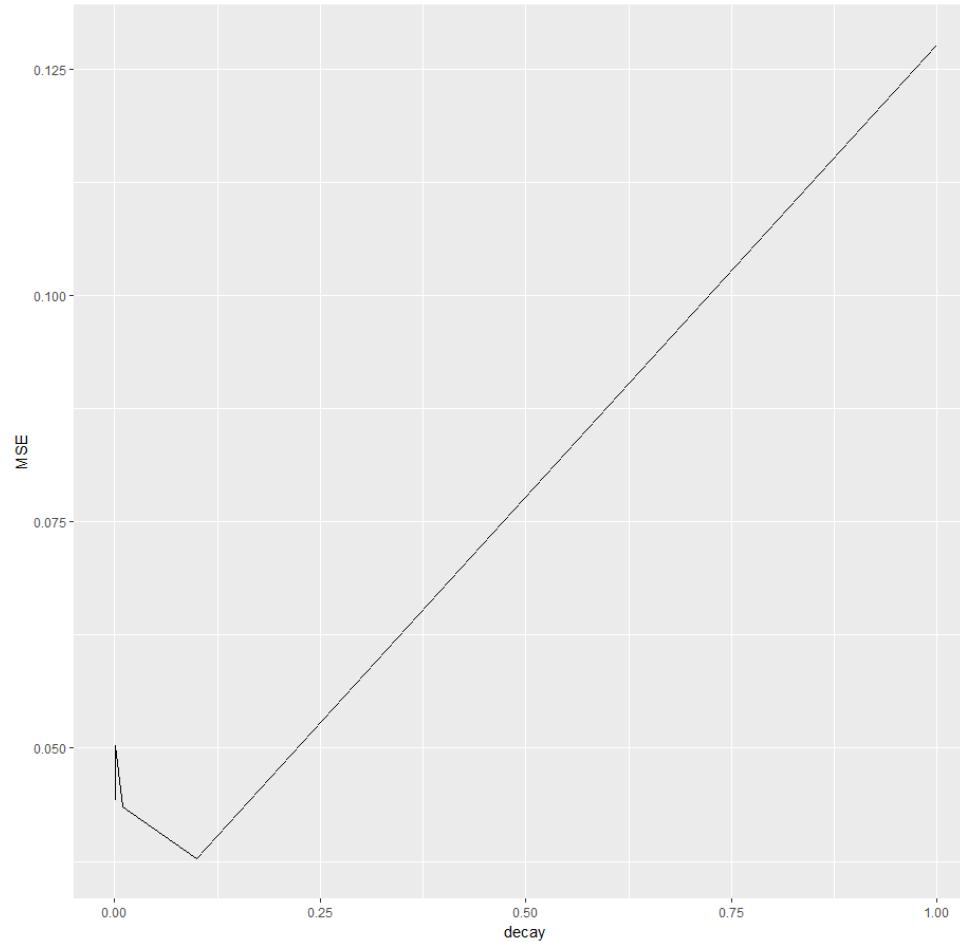


Figure 4. Schematic diagram of the model with selection based on problem features and variable performance criteria.

autoML is very hard



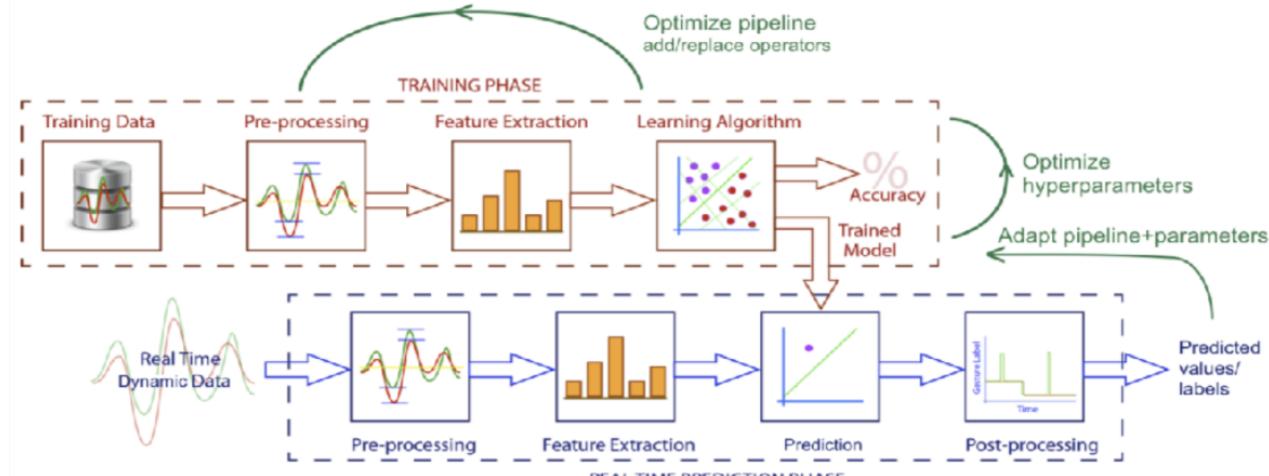
thanks Catarina Félix!

ML

24

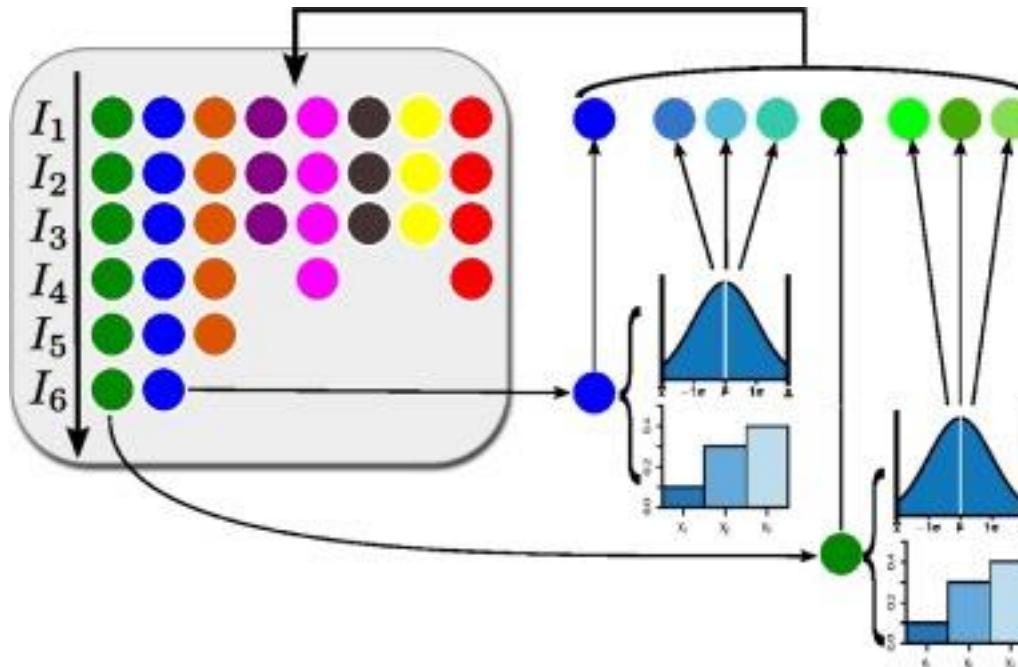
autoML is extremely hard!

AUTOMATING MACHINE LEARNING PIPELINES



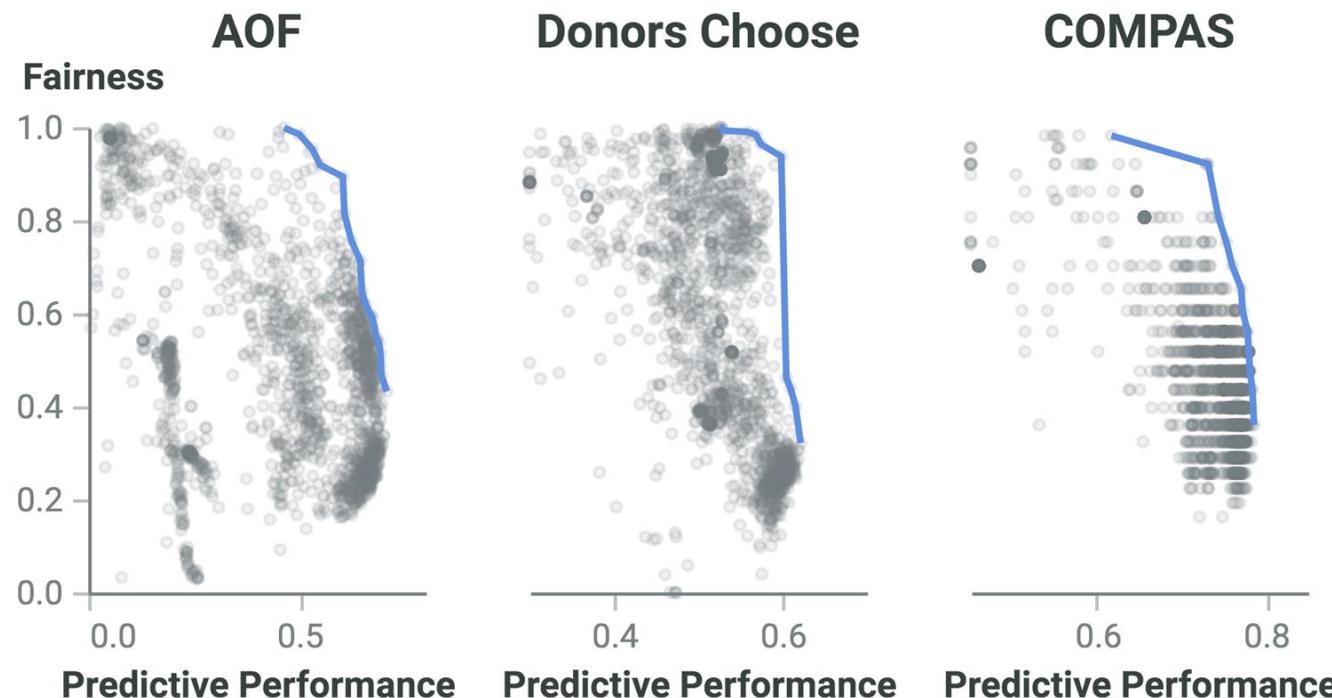
can't resist it: his slides look so much better than mine!

trendy autoML approaches: search and metalearning



and it's not only about predictive performance!

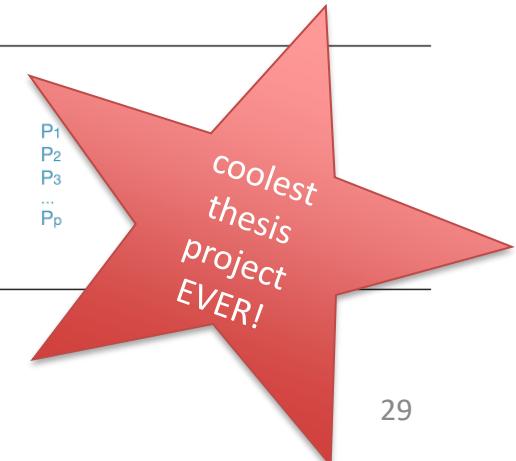
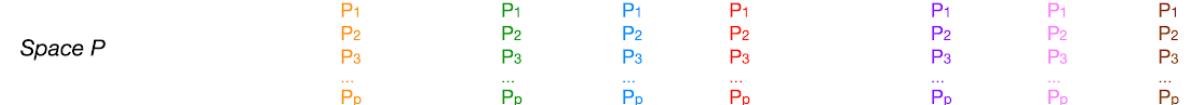
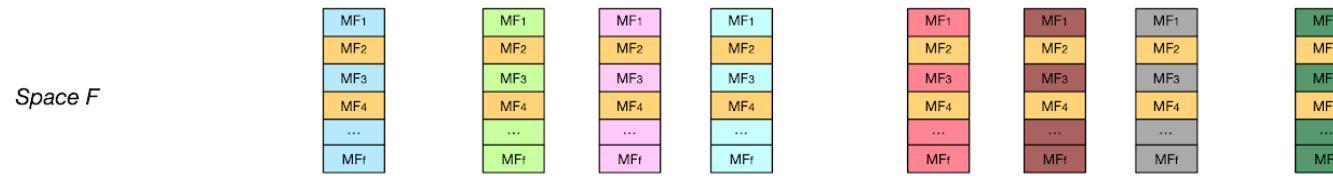
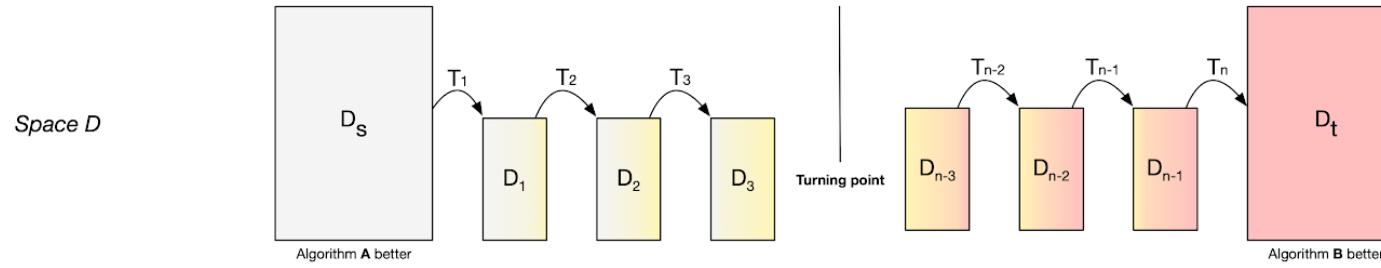
- responsible AI
 - is *accuracy vs fairness* a real problem?
 - i.e. if I want to promote fair models, I have to sacrifice predictive performance



todo

- the world where automated ml lives
 - a world of **many models**
 - needs **model management**
 - metalearning/automl can help
 - but **opportunities and challenges** are still open

coolest metalearning ever (1/2): dataset morphing to understand ML algorithm behavior



coolest metalearning ever (2/2): dataset characterisation

```
> iris
```

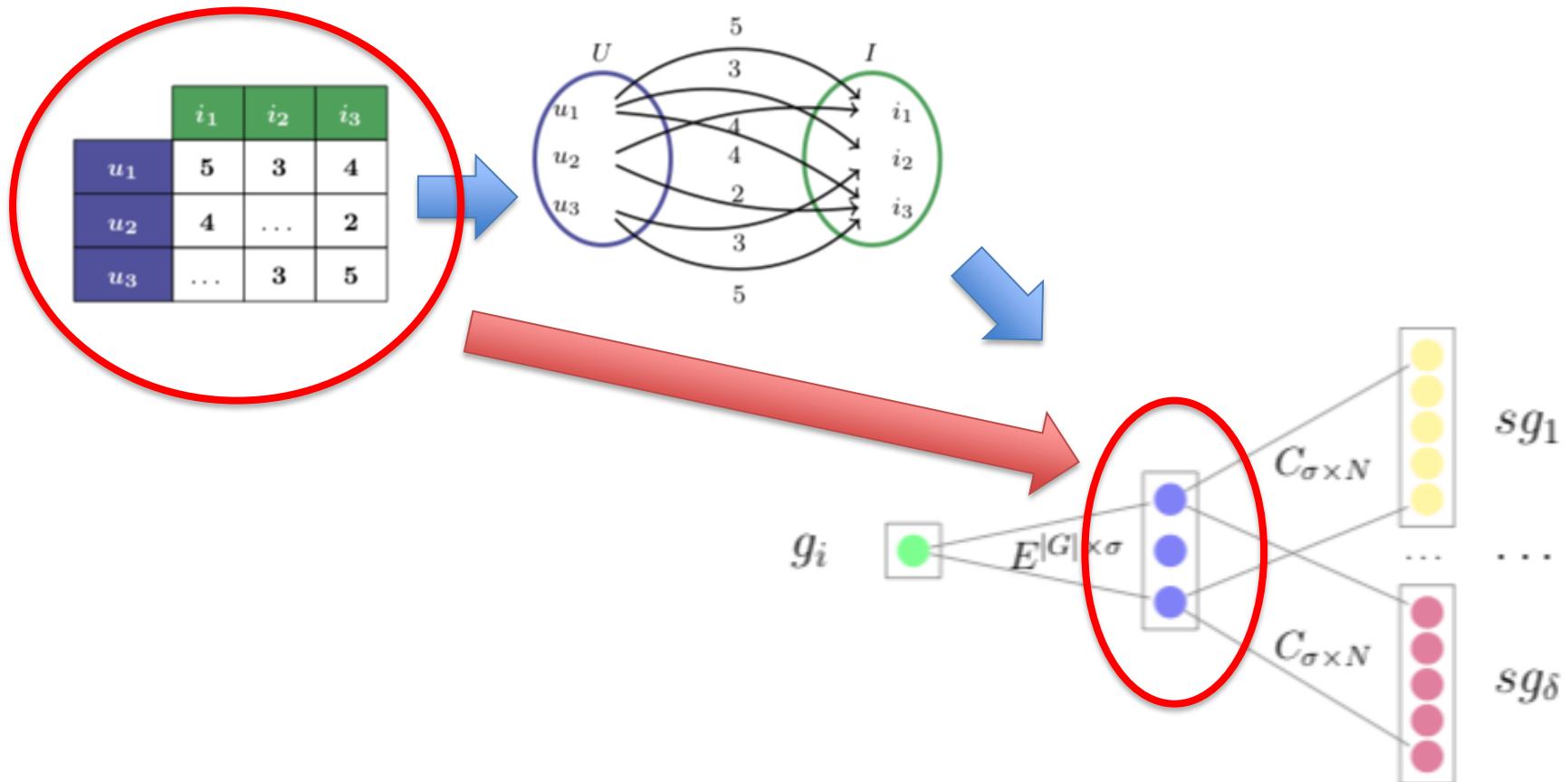
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa



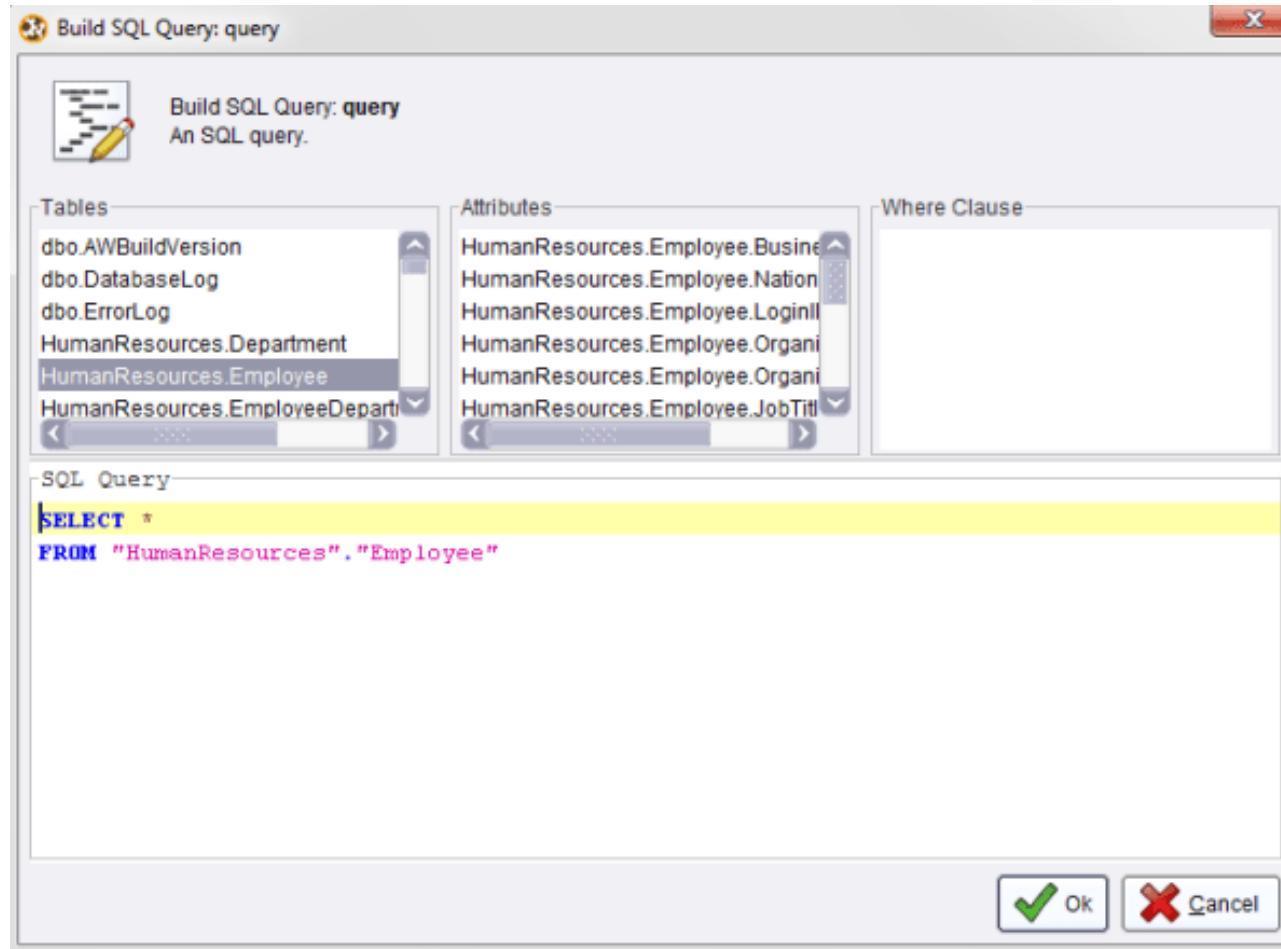
meta-data

	n.examples	n.attributes	n.classes	def.accuracy	meta.target
imports_85	205	25	0	0.3268293	DT
ionosphere	351	33	0	0.6410256	LD
iris	150	4	0	0.3333333	DT
kr-vs-kp	3196	36	0	0.5222153	DT
lung-cancer	32	56	0	0.4062500	LR

coolest metalearning ever (2/2): dataset embeddings



but what is fundamentally wrong?



how to make it fundamentally right?

```
CREATE TABLE LoanRequest (
    ID INT PRIMARY KEY,
    CustomerID INT REFERENCES Customer(ID) NOT
NULL,
    Value INT,
    Default ENUM('y','n') TARGET,
);
```

```
SELECT ID, PREDICT(Default) FROM LoanRequest;
```

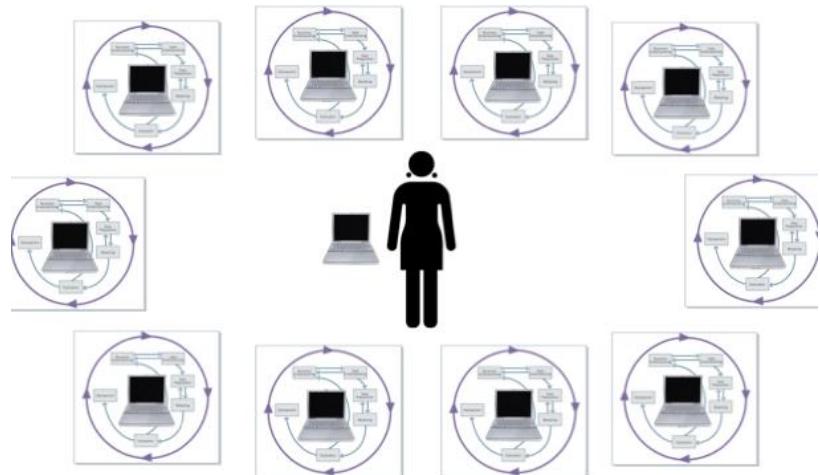
the way to a (real) empirical science of machine learning

- metalearning to understand the behavior of algorithms
 - eg when to do resampling in unbalanced datasets

Coverage	LWNorm ($\times 10^{-2}$)	Ranking	Conditions
No preproc.			
21	3.3992	j>c>d>e>bf>i>a>h>g	<i>statistical_kurtosis</i> >= 17.9168
21	2.8934	j>b>d>e>f>c>i>a>g>h	<i>statistical_cov</i> <= 0.0234
21	2.7911	j>b>d>c>f>e>a>i>g>h	<i>statistical_eigenvalues</i> <= 0.2581
21	2.7911	j>b>d>c>f>e>a>i>g>h	<i>statistical_var</i> <= 0.2581
42	2.3839	j>c>d>ef>b>i>h>a>g	<i>general_nr_inst</i> >= 376.0
21	2.8448	j>c>e>d>f>b>h>i>g>a	<i>complexity_n1</i> <= 0.0675
41	2.8327	j>c>e>d>b>f>h>i>g>a	<i>complexity_l2</i> <= 0.0421
24	2.6046	j>c>d>e>f>b>i>h>a>g	<i>complexity_t3</i> <= 0.0031
41	2.2675	j>c>e>d>b>h>f>gi>a	<i>complexity_n4</i> <= 0.0611
21	2.4585	j>c>d>e>f>b>h>i>g>a	<i>typology_border</i> <= 0.0858
41	2.2161	j>c>d>b>e>f>h>i>g>a	<i>typology_safe</i> >= 0.5334
41	2.9260	j>c>e>d>b>f>h>i>g>a	<i>landmarking_linear_discr</i> >= 0.9225
21	2.7286	j>e>c>d>b>f>h>i>g>a	<i>landmarking_nn</i> >= 0.9750
41	2.7001	j>c>e>d>b>f>h>i>a>g	<i>landmarking_nn</i> >= 0.9052
Do preproc.			
21	3.0865	h>c>d>f>ae>i>g>b>j	<i>statistical_kurtosis</i> <= -1.3063
21	2.2963	h>a>ci>b>d>f>j>e>g	<i>statistical_sparsity</i> >= 0.4085
21	2.7049	h>c>bg>i>d>f>a>e>j	<i>typology_border</i> >= 0.6555
21	2.4990	h>a>d>b>c>e>g>fi>j	<i>complexity_t3</i> >= 0.0668
22	2.2101	h>a>g>c>f>e>d>i>b>j	<i>complexity_t2</i> >= 0.1250
41	2.1513	c>h>f>b>a>d>e>i>j>g	<i>complexity_f3</i> >= 0.9831
41	2.1513	c>h>f>b>a>d>e>i>j>g	<i>complexity_f4</i> >= 0.9831
21	2.3544	h>c>b>a>f>d>ei>g>j	<i>landmarking_elite_nn</i> <= 0.5788
21	2.9550	h>c>a>b>f>dg>i>e>j	<i>landmarking_best_node</i> <= 0.6557

wrap-up

- model management
 - exciting field
 - e.g. autoML (<http://www.automl.org/>)
 - new challenges
- do not forget the basic issues
 - ... not all of them, at least
- learn from other areas
 - e.g., algorithm portfolios



Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection.
ACM Comput. Surv. 2008;41(1):1-25

acknowledgements

- Supervisors: P. Brazdil (UP), J.P. Costa (UP)
- Colleagues: METAL project, A. Carvalho (USP), R. Prudêncio (UFPE), C. Giraud-Carrier (BYU), R. Vilalta (UT), P. Flach (UB), H. Ferreira (UP)
- Students: P. Abreu (UP), C. Félix (UP), C. Gomes (UP), F. Pinto (UP), M. Nozari (UP), T. Cunha (UP), T. Gomes (UFPE), J. Kanda (USP), T. Lucas (UFPE), P. Miranda (UFPE), E. Partodikromo (UL), F. Pinto (UP), C. Rebelo (UP), A. Rossi (USP), B. Souza (USP)

“THE” Book

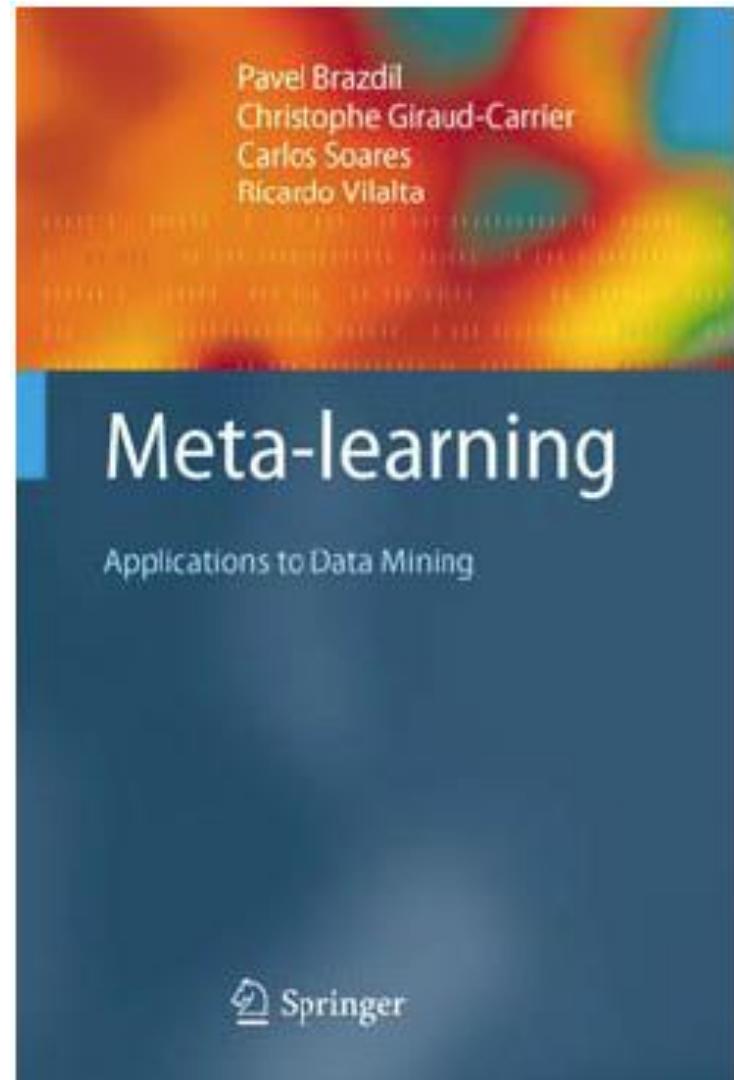
Metalearning – Applications to Data Mining

Pavel Brazdil

Christophe Giraud-Carrier

Carlos Soares

Ricardo Vilalta



<http://www.springer.com/computer/artificial/book/978-3-540-73262-4>

another book (not so interesting... ;-))



Home Blog AutoML ▾ AAD ▾ Analysis ▾ **Book** Jobs Events Team & Partners ▾

AUTOML: METHODS, SYSTEMS, CHALLENGES (NEW BOOK)

Editors: Frank Hutter, Lars Kotthoff, Joaquin Vanschoren

We're in the process of finishing this edited book, and it will be ready for sale by NIPS 2018. Next to publishing, we will keep the book open access. Below, we share preliminary versions of the chapters; at this point in time, **these are all drafts, before copy editing**.

Part 1: AutoML Methods

This part comprises highly up-to-date overview chapters on the common foundations behind all AutoML systems.

Chapter 1: Hyperparameter Optimization. By Matthias Feurer and Frank Hutter

Chapter 2: Meta Learning. By Joaquin Vanschoren

Chapter 3: Neural Architecture Search. By Thomas Elsken, Jan-Hendrik Metzen and Frank Hutter

Part 2: AutoML Systems

This part comprises in-depth descriptions of a broad range of available AutoML systems that can be used for effective machine learning out of the box.

Chapter 4: Auto-WEKA. By Lars Kotthoff and Chris Thornton and Holger H. Hoos and Frank Hutter and Kevin Leyton-Brown