IP ComVICS
Le Creusot, 16 June 2014

Nuno Escudeiro, nfe@isep.ipp.pt

OUTLINE

INTRODUCTION

MOTIVATIONS

Data mining

- finding valuable patterns in data
- data mining methods expect structured datasets

Text mining – data mining over natural language textual data

- big data
- availability of text in digital form

- high volume vs high quality
- Natural Language Processing (NLP) → hand-made rules, scalability issues

isep instituto superior de engenharia do porto

MOTIVATIONS

Data mining: finding valuable patterns in data

- big data
- availability of text in digital form

- high volume vs high quality
- NLP, hand-made rules, scalability

Data mining methods expect highly structured datasets

MOTIV

Data ta

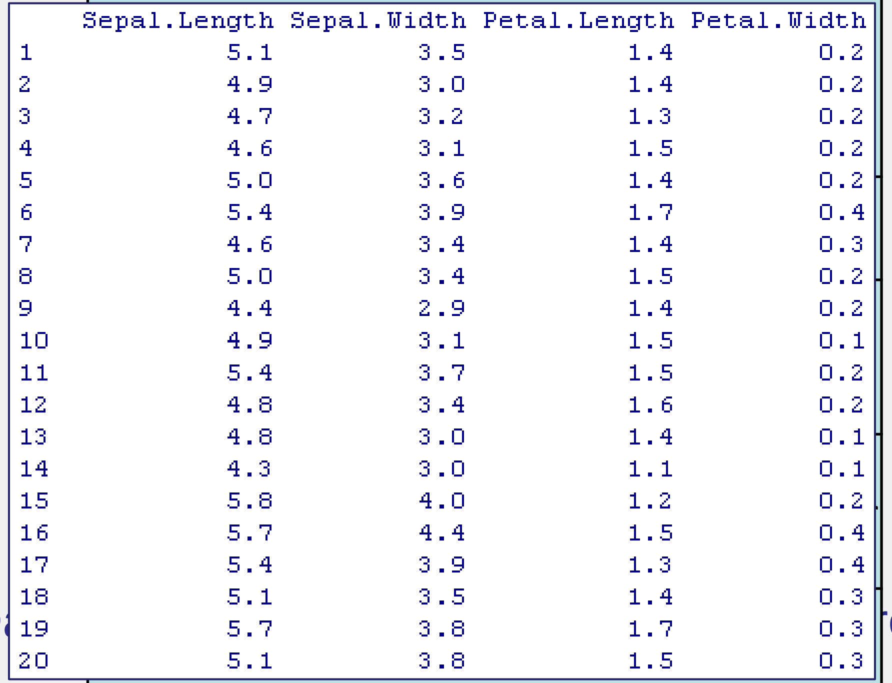
Data red datasets

Traditional data mining
structured (tabular form)
numbers
understandable by domain specialists
descriptive attributes are permanent, known in advance and may be based on prior design
all examples are described by the same set of attributes
numerical (greater than, less than) / categorical (list of unordered codes) attributes
all the possible values of a given attribute are known in advance and recorded uniformly for every example
semantic of attributes is invariant
adding a new example is adding a new row with a vector of measures for each of the attributes
examples are rows, datasets are matrices, mathematical methods are aplicable

MOTIVA

Data ta

Traditional data mining
structured (tabular form)
numbers
understandable by domain specialists
adding a new example is adding a new row with a vector of measures for each of the attributes
examples are rows, datasets are matrices, mathematical methods are aplicable



```
   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
6           5.4         3.9          1.7         0.4
7           4.6         3.4          1.4         0.3
8           5.0         3.4          1.5         0.2
9           4.4         2.9          1.4         0.2
10          4.9         3.1          1.5         0.1
11          5.4         3.7          1.5         0.2
12          4.8         3.4          1.6         0.2
13          4.8         3.0          1.4         0.1
14          4.3         3.0          1.1         0.1
15          5.8         4.0          1.2         0.2
16          5.7         4.4          1.5         0.4
17          5.4         3.9          1.3         0.4
18          5.1         3.5          1.4         0.3
19          5.7         3.8          1.7         0.3
20          5.1         3.8          1.5         0.3
```

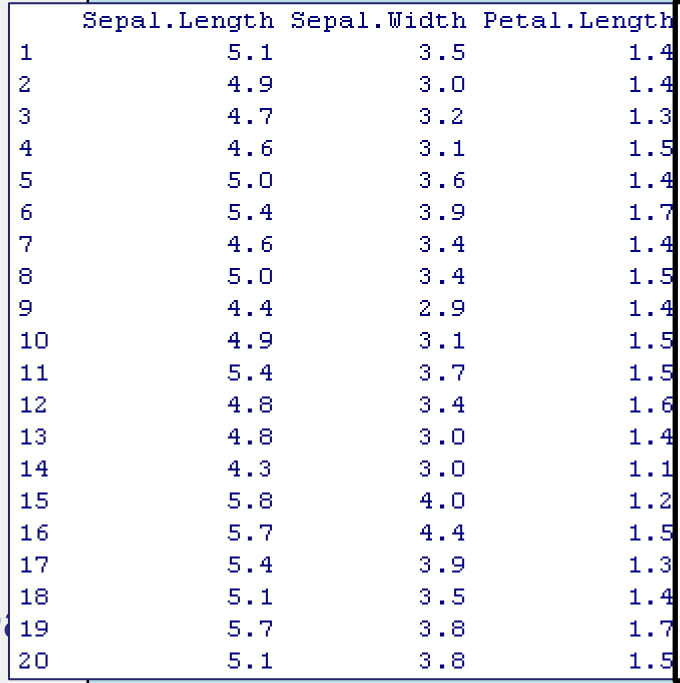
D red datasets

MOTIVA

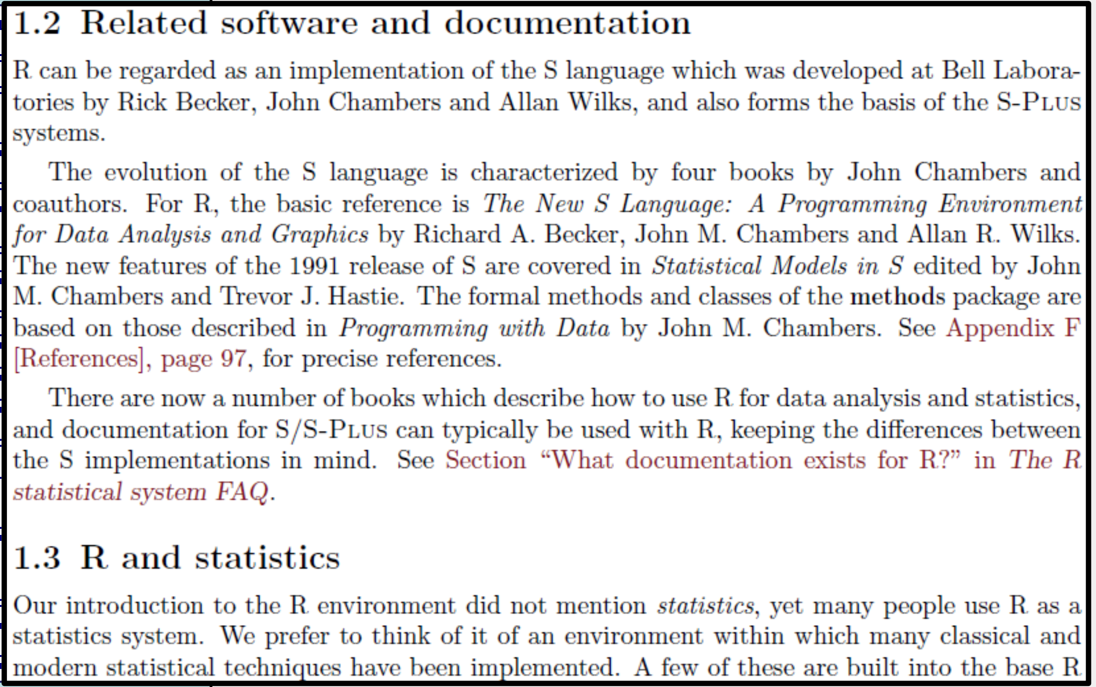
Data ... ta

Da

Traditional data mining
structured (tabular form)
numbers
understandable by domain specialists
adding a new example is adding a new row with a vector of measures for each of the attributes
examples are rows, datasets are matrices, mathematical methods are aplicable



```
    Sepal.Length Sepal.Width Petal.Length
1            5.1         3.5          1.4
2            4.9         3.0          1.4
3            4.7         3.2          1.3
4            4.6         3.1          1.5
5            5.0         3.6          1.4
6            5.4         3.9          1.7
7            4.6         3.4          1.4
8            5.0         3.4          1.5
9            4.4         2.9          1.4
10           4.9         3.1          1.5
11           5.4         3.7          1.5
12           4.8         3.4          1.6
13           4.8         3.0          1.4
14           4.3         3.0          1.1
15           5.8         4.0          1.2
16           5.7         4.4          1.5
17           5.4         3.9          1.3
18           5.1         3.5          1.4
19           5.7         3.8          1.7
20           5.1         3.8          1.5
```



1.2 Related software and documentation

R can be regarded as an implementation of the S language which was developed at Bell Laboratories by Rick Becker, John Chambers and Allan Wilks, and also forms the basis of the S-Plus systems.

The evolution of the S language is characterized by four books by John Chambers and coauthors. For R, the basic reference is *The New S Language: A Programming Environment for Data Analysis and Graphics* by Richard A. Becker, John M. Chambers and Allan R. Wilks. The new features of the 1991 release of S are covered in *Statistical Models in S* edited by John M. Chambers and Trevor J. Hastie. The formal methods and classes of the **methods** package are based on those described in *Programming with Data* by John M. Chambers. See Appendix F [References], page 97, for precise references.

There are now a number of books which describe how to use R for data analysis and statistics, and documentation for S/S-Plus can typically be used with R, keeping the differences between the S implementations in mind. See Section "What documentation exists for R?" in *The R statistical system FAQ*.

1.3 R and statistics

Our introduction to the R environment did not mention *statistics*, yet many people use R as a statistics system. We prefer to think of it of an environment within which many classical and modern statistical techniques have been implemented. A few of these are built into the base R

MOTIVA

Data

Data

Traditional data mining	Text mining
structured (tabular form)	non-structured (free form)
numbers	text
understandable by domain specialists	readable, meaningful to non-specialists
descriptive attributes are permanent, known in advance and may be based on prior design	descriptive attributes of a text are the words in it (alone?, in pairs?), document structure, letter case, letter format, metadata (in which case we turn to the classical paradigm)
all examples are described by the same set of attributes	
numerical (greater than, less than) / categorical (list of unordered codes) attributes	
all the possible values of a given attribute are known in advance and recorded uniformly for every example	
semantic of attributes is invariant	semantics depends on linguistic matters such as context and grammar
adding a new example is adding a new row with a vector of measures for each of the attributes	
examples are rows, datasets are matrices, mathematical methods are aplicable	

isep instituto superior de engenharia do porto

MOTIVATIONS

Natural Language Processing (NLP)

- linguistic approach
- hand-written rules
- theasaurus, lexical databases (e.g. WordNet - http://wordnet.princeton.edu/)
- demands for specialists in linguistics
- some applications of NLP

  i) question answering

  ii) automatic translation

  iii) text summarization

  iv) sentence parsing

Statistical methods perform well on these problems!

FROM TEXT TO NUMBERS

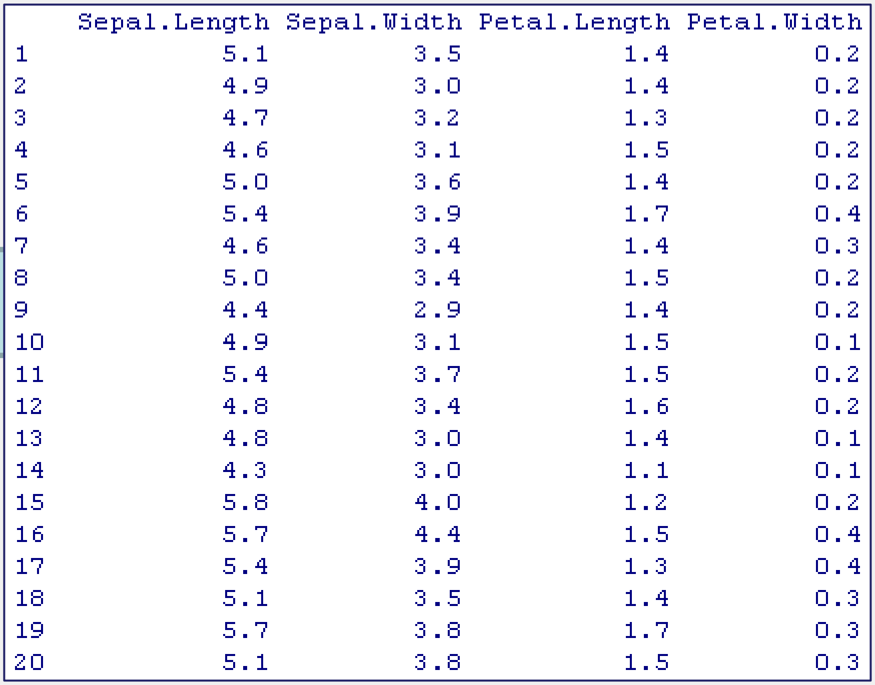
Each word is a descriptive attribute

Word frequency

To be or not to be

Sparse matrices

High number of attributes (easily reach 10^3 - 10^5)



```
   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
6           5.4         3.9          1.7         0.4
7           4.6         3.4          1.4         0.3
8           5.0         3.4          1.5         0.2
9           4.4         2.9          1.4         0.2
10          4.9         3.1          1.5         0.1
11          5.4         3.7          1.5         0.2
12          4.8         3.4          1.6         0.2
13          4.8         3.0          1.4         0.1
14          4.3         3.0          1.1         0.1
15          5.8         4.0          1.2         0.2
16          5.7         4.4          1.5         0.4
17          5.4         3.9          1.3         0.4
18          5.1         3.5          1.4         0.3
19          5.7         3.8          1.7         0.3
20          5.1         3.8          1.5         0.3
```

FROM TEXT TO NUMBERS

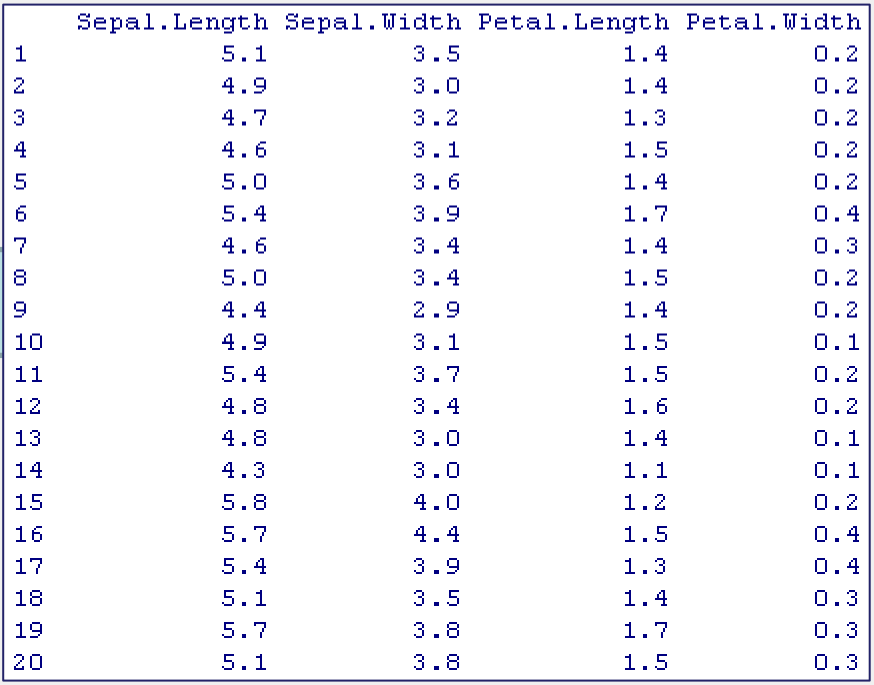
Each word is a descriptive attribute

Word frequency

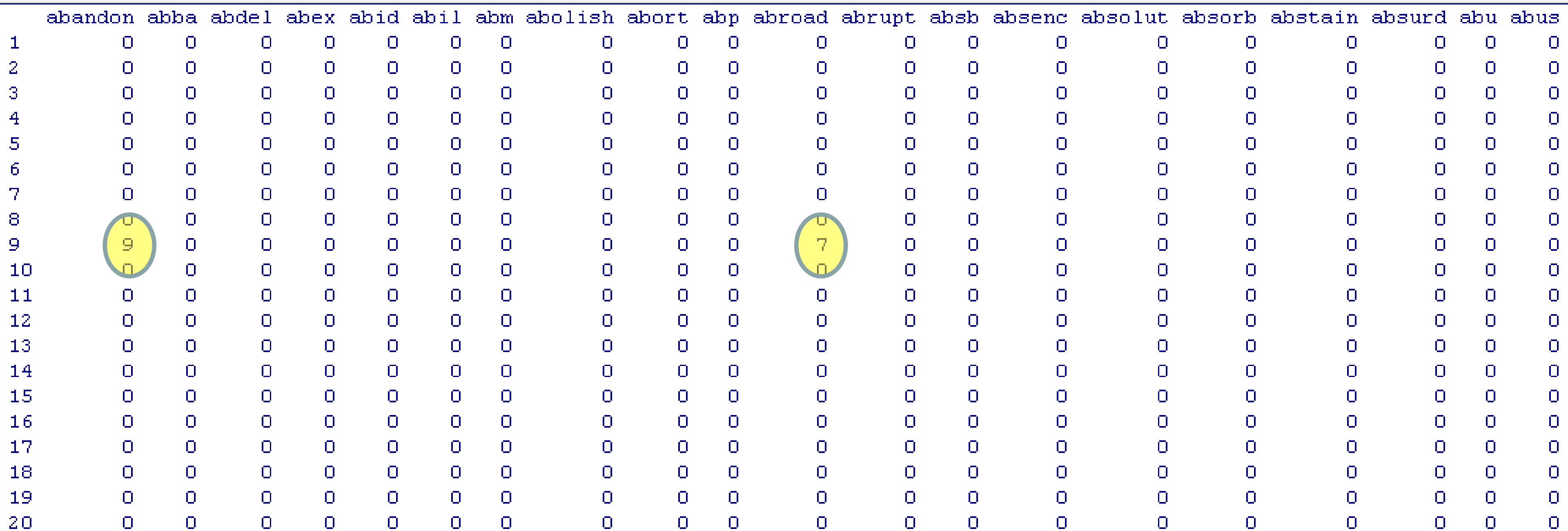
To be or not to be

Sparse matrices

High number of attributes (easily reach 10^3 - 10^5)



```
   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
6           5.4         3.9          1.7         0.4
7           4.6         3.4          1.4         0.3
8           5.0         3.4          1.5         0.2
9           4.4         2.9          1.4         0.2
10          4.9         3.1          1.5         0.1
11          5.4         3.7          1.5         0.2
12          4.8         3.4          1.6         0.2
13          4.8         3.0          1.4         0.1
14          4.3         3.0          1.1         0.1
15          5.8         4.0          1.2         0.2
16          5.7         4.4          1.5         0.4
17          5.4         3.9          1.3         0.4
18          5.1         3.5          1.4         0.3
19          5.7         3.8          1.7         0.3
20          5.1         3.8          1.5         0.3
```



```
   abandon abba abdel abex abid abil abm abolish abort abp abroad abrupt absb absenc absolut absorb abstain absurd abu abus
1        0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
2        0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
3        0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
4        0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
5        0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
6        0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
7        0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
8        0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
9        9    0     0    0    0    0   0       0     0   0      7      0    0      0       0      0       0      0   0    0
10       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
11       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
12       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
13       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
14       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
15       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
16       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
17       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
18       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
19       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
20       0    0     0    0    0    0   0       0     0   0      0      0    0      0       0      0       0      0   0    0
```

APPLICATIONS

- Document classification (text categorization)

- Clustering and organizing a collection of documents (corpus)

- Information retrieval (document matching, search engines, document similarity)

- Information extraction (NER, summarization)

TEXT MODELING

BAG-OF-WORD MODELS

- **Boolean**
- **Vector space** (assumes independence of index terms)
  - i) TF (Term Frequency)
  - ii) TFxIDF (Term Frequency x Inverse Document Frequency)
  - iii) Entropy weighting
- **Generalized vector space** (co-occurence induces dependency)
- **N-grams** (index terms are groups of n-words or n-chars rather than single words)

isep instituto superior de engenharia do porto

STRUCTURED MODELS (mainly used for IR)

- **Non-overlapping list model** (lists of non-overlapping text regions)
- **Proximal nodes model** (set of independent hierarchical indexing structures)
- **Path prefixing** (hypertext, prefixing words/phrases with their html tags)
- **Relational model** (represents corpora in a relational schema including relations like

  ContainsText(domNode, text)

  PartOf(domNode, domNode)

  LinksTo(scrDomNode, dstDomNode)

TEXT PROCESSING

TEXT PROCESSING

a) Pre-processing

The pre-processing stage comprises the tasks that are required to obtain a suitable document representation, valid for the subsequent automatic learning phase. This stage includes text preparation and text representation.

b) Learning

Infering knowledge using adequate machine learning algorithms depending on the task at hand.

c) Generalizing

Applying the infered model to new examples.

d) Presenting

Presenting results.

PRE-PROCESSING STAGE:

Text preparation: takes a text document as input and returns a set of features describing it. This phase includes several steps that attempt to eliminate non-informative features and might involve some or all of the following tasks´:

Tokenization – breaking the text document into tokens, commonly words; includes all sorts of lexical analysis steps, such as: eliminating punctuation, eventually numbers, accents and extra spacing, converting to lower/upper case

Stop-word removal – removing irrelevant terms; requires a list of stop-words (words to eliminate)

Stemming or lemmatization – reducing words to their semantic root; the Porter algorithm

Text representation: the full process of selecting features and computing their weights is known as *indexing*.

Feature selection – defining index terms, words, n-grams, …: the features that will be used for document modeling (vocabulary)

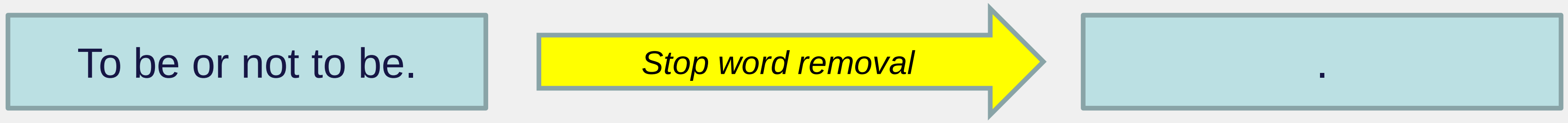
Computing index **term weight**

PRE-PROCESSING STAGE:

The tasks performed during the **pre-processing** stage **depend on the model** that will be used to represent text.
The application of these pre-processing tasks must be carefully done because the predictive power of words is highly dependent on the topic of interest.

Another essential aspect to consider is the **language** in which the document is written, which determines, at least, the list of **stop-words** and the **stemming algorithm** to use.

Stop-words removal is controversial. Removing words from a text, even those that in a linguistic sense have low semantic value, always reduces the information contained in the text document. Stop-word removal reduces the dimensionality of the feature space at a cost of loosing some information. A compromise solution must be set so that this information loss does not becomes counterproductive.



PRE-PROCESSING STAGE:

D1: A student is a learner, or someone who attends an educational institution.
D2: The average age of higher education students is 29.

Tokenization:
D1: a student is a learner or someone who attends an educational institution
D2: the average age of higher education students is

Stop-word removal:
D1: student learner someone who attends educational institution
D2: average age higher education students

Stemming:
D1: student learn someone who attend education institution
D2: average age higher education student

Indexing:

	age	attend	average	education	higher	institution	learn	someone	student	who
D1	0	1	0	1	0	1	1	1	1	1
D2	1	0	1	1	1	0	0	0	1	0

PRE-PROCESSING STAGE:

D1: A student is a learner, or someone who attends an educational institution.
D2: The average age of higher education students is 29.

Tokenization:
D1: a student is a learner or someone who attend educational institution
D2: the average age of higher educatio

Stop-word removal:
D1: student learner someone wh
D2: average age higher educati

Stemming:
D1: student learn someone who attend
D2: average age higher education student

Which words would you remove from the vocabulary below if the purpose is to distinguish between D1 and D2?

Indexing:

	age	attend	average	education	higher	institution	learn	someone	student	who
D1	0	1	0	1	0	1	1	1	1	1
D2	1	0	1	1	1	0	0	0	1	0

isep instituto superior de engenharia do porto

VECTOR MODEL – TFXIDF:

- the **term frequency** factor, TF, a measure of intra-cluster similarity; computed as the number of times that the term occurs in the document, normalized in a way as to make it independent of document length and

- the **inverse document frequency** factor, IDF, a measure of inter-cluster dissimilarity; weighs each term according to its discriminative power in the entire collection.

$$w_{ij} = TF(d_i, t_j) \cdot IDF(t_j)$$

$$w_{ij} = f_{ij} \cdot log(\frac{N}{N_j})$$

VECTOR MODEL – TFXIDF:

- the **term frequency** factor, TF, a measure of i ted as the number of times that the term occu way as to make it independent of docu
- the **inverse document frequency** fac dissimilarity; weighs each term accordin collection.

What is the TFxIDF weight of a term that appears in all document in the corpus?

$$w_{ij} = TF(d_i, t_j) \cdot IDF(t_j)$$

$$w_{ij} = f_{ij} \cdot log(\frac{N}{N_j})$$

VECTOR MODEL – COSINE SIMILARITY:

Q: higher education student age

D1: A student is a learner, or someone who attends an educational institution.
D2: The average age of higher education students is 29.

	age	attend	average	education	higher	institution	learn	someone	student	who
D1	0	1	0	1	0	1	1	1	1	1
D2	1	0	1	1	1	0	0	0	1	0
Q	1	0	0	1	1	0	0	0	1	0

$cos(D_i, Q)$	Q
D_1	0.38
D_2	0.89

$$sim(d_i, d_k) = cosine(d_i, d_k) = \frac{\vec{d_i} \otimes \vec{d_k}}{\left\|\vec{d_i}\right\| \cdot \left\|\vec{d_k}\right\|} = \frac{\sum_{j=1}^{t} w_{ij} w_{kj}}{\sqrt{\sum_{j=1}^{t} w_{ij}^2} \sqrt{\sum_{j=1}^{t} w_{kj}^2}}$$

TEXT PROCESSING FUNCTIONS

- **Part-Of-Speech (POS) tagging** (gramatical class of words: verb, noun, adverb, …)
- **Named Entity Recognition (NER)** (recognition of types of noun phrases: persons, places, organizations, dates, …)
- **Parsing** (finding the relations of a single word in a sentence to all others and its function: subject, object, …)
- **Word sense disambiguation** (semantics, identify word meaning)
- **Phrase recognition** (group individual tokens into units)

EVALUATION

- **Precision**

  Defined as the ratio between the number of documents correctly classified and the total number of documents classified in the category

- **Recall**

  Defined as the ratio between the number of documents correctly classified and the total number of documents in the category

- **F-measure**

  The F-measure combines recall and precision in a unique indicator. F1 gives the same weight to precison and recall.

- **Accuracy, Error**

  Complementary measures of the error probability of the classifier given a certain category. Accuracy is defined as the ratio of the number of correctly classified examples by the total number of evaluated examples, while error is defined as the ratio of the number of incorrectly classified examples by the total number of evaluated examples.

EVALUATION

- **Precision**

  Defined as the ratio between the number of documents correctly classified and the total number of documents classified in the category

- **Recall**

  Defined as the ratio between the number of documents correctly classified and the total number of documents in the category

- **F-measure**

  The F-measure combines
  gives the same weight to p

$$F_\beta = \frac{(\beta^2 + 1) \times precision \times recall}{\beta^2 \times precision + recall}$$

- **Accuracy, Error**

  Complementary measures of the error probability of the classifier given a certain category. Accuracy is defined as the ratio of the number of correctly classified examples by the total number of evaluated examples, while error is defined as the ratio of the number of incorrectly classified examples by the total number of evaluated examples.

EVALUATION

These measures are defined for **binary classifiers**. To measure performance in **multi-class problems** there are two common approaches that aggregate the measures evaluated at each singular class: macro-averaging and micro-averaging.

Macro-averaged measures are obtained by first computing the individual scores, for each individual class, and then, averaging these scores to obtain the global measure.

Micro-averaging measures are obtained by first computing the individual scores of each document and then computing their average.

There is an important distinction between these two forms of aggregation : macro-averaging gives equal weights to all classes while micro-averaging gives equal weight to every document.

- **Micro-average**

  Same weight to all documents
- **Macro average**

  Same weight to all categories

What is the impact of micro/macro averages when in presence of imbalanced class distributions?

TOOLS FOR TEXT MINING

R, tm package – http://www.r-project.org/

Mallet – http://mallet.cs.umass.edu/

Lucene – http://lucene.apache.org/

OpenNLP – http://opennlp.apache.org/

RapidMiner – http://rapid-i.com/

R LANGUAGE, TM PACKAGE

R is an integrated suite of software facilities for data manipulation, calculation and graphical display.
Among other things it has an effective data handling and storage facility, a suite of operators for calculations on arrays, in particular matrices, a large, coherent, integrated collection of intermediate tools for data analysis, graphical facilities for data analysis and display and simple and effective programming language which includes: conditionals, loops, user defined recursive functions and input and output facilities.

R prompt

```
>
```

Quitting R

```
> q()
```

Help

```
> ?command or >help(command)
```

R is **case sensitive**

Comments

```
# content ignored up to the next newline
```

Char constant

```
"this is a char constant"
```

Elementary commands, separated by '`;`' can be grouped together into one compound expression by braces ('`{`' and '`}`').

Some useful commands:

```
> ls()
> rm()
```

Assignment operator:

```
> x <- 3 #local
> x <<- 3 #global
```

Coercion, change of mode, casting

```
as.numeric(), as.character(), as.matrix(), ...
```

R operates on named data structures. The simplest such structure is the numeric vector, which is a single entity consisting of an ordered collection of numbers.

```
    > x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```

Vector arithmetic

```
    > v <- 2*x + y + 1
```

Elementary arithmetic operators:

```
+    -    *    /    ^
```

Arithmetic functions:

```
    log(), exp(), sin(), cos(), sqrt(), sum(x), max(x), min(x),
    length(x)
```

Indexing:

```
    > x <- c(43, 67, 23, 12, 87)
    > x
    [1] 43 67 23 12 87
    > x[2]
    [1] 67
```

isep instituto superior de engenharia do porto

Other types of objects (besides **vectors**):

- **matrices** or more generally arrays are multi-dimensional generalizations of vectors.

```
> m[1,1]
> m[1,]
```

- **factors** provide compact ways to handle categorical data. A factor is a vector object used to specify a discrete classification (grouping) of the components of other vectors of the same length.

- **lists** are a general form of vector in which the various elements need not be of the same type, and are often themselves vectors or lists.

```
> x <- list()
> x[[1]] <- c(1,2,3)
```

- **data frames** are matrix-like structures, in which the columns can be of different types

```
> planets <- data.frame(name=planets, temperature=tc)
> planets[,1] or planets$name or planets[,"name"]
```

- **functions** are themselves objects in R which can be arguments to other functions

```
> d <- function(x, y) {sqrt(sum((x-y)^2))}
> d(0,c(1,1,1))
> [1] 1.732051
```

Control statements: IF

```
if (expr_1) expr_2 else expr_3
```

```
if(i == 1) {
    sed.ks[i] <- NA
} else {
    sed.ks[i] <- f.stop.sedks(prev.post, post, udIdx)
}
```

Control statements: FOR

```
for (name in expr_1) expr_2
```

```
for(j in 1:nrow(ad)) {
     for(k in 1:length(itkc)) {
             ipd[j,k] <- post[uIdx[j], itkc[k]] / ad[j,k]
     }
}
```

Control statements: WHILE

```
while (condition) expr

  while (length(unlabeledIdx) != 0) {
       labeledIdx <- union(labeledIdx, query[i])
       unlabeledIdx <- setdiff(trainingIndex, labeledIdx)
       i <- i + 1
  }
```

isep instituto superior de engenharia do porto

R PACKAGES

All R functions and datasets are stored in packages. Only when a package is loaded are its contents available.

To see which packages are installed at your site, issue the command:

```
> library()
```

To install packages:

```
> install.packages()
```

To load a particular package (e.g., the text mining package), use:

```
> library(tm)
```

or

```
> require(tm)
```

To see which packages are currently loaded, use:

```
> search()
```

R-intro

isep instituto superior de engenharia do porto

TEXT PROCESSING: tm PACKAGE

A modular, extensible, thoroughly designed text mining infrastructure for R.

- **Pre-processing**: data import, stemming, stopword removal, part of speech tagging, synonyms, . . .
- Basis **analysis techniques**: count based evaluation, text clustering, text classification, . . .
- Access to more **advanced functionality**: full integration with string kernels, latent semantic analysis, . . .
- **Export** of term-document matrices: basically all methods in R working on matrices

TEXT PROCESSING: tm PACKAGE

```
> removeNumbers()
> removePunctuation()
> tokenizer()
> stemDocument()

> Dictionary()
> DocumentTermMatrix()
> termFreq()
> tm_intersect()
> findFreqTerms()
> removeSparseTerms()
…
```

tm reference

isep instituto superior de engenharia do porto

information
scorecards
analyze
text mining
metrics
applications
objectives
capitaliz
manage

CASE STUDY

```
# load corpus
reviews <- Corpus(DirSource("./reviews"), readerControl = list(reader =
    readPlain, language = "english", load = TRUE))

# generate vector space model, computing index term’s weights TF
dtm <- DocumentTermMatrix(reuters)

# CHECK IT
inspect(reviews)
inspect(reviews[1])

str(dtm)
colnames(dtm) # YOU
```

HANDS-ON TEXT MINING

R PACKAGES

1. Install R

2. Install required libraries

   `tm` -- text mining
   `SnowballC` -- stemming
   `RWeka` -- N-gram tokenizer

   `e1071` -- svm
   `class` -- knn

isep
instituto superior de engenharia do porto

THE CONTEST

1. Get a review, guess the game
2. Building a text classifier

Everything is valid:

- Stop word removal
- Removal of non-useful words (low frequency/high frequency)
- Re-weighting
- Stemming
- POS tagging
- N-grams
- Correlation between terms and topics
- More data, Less data
- Use of Wordnet
- Other classification algorithms (neural networks, decison trees, …)
- …

https://www.dropbox.com/sh/zeb9iqiy9u0isdx/_EhjMTTRXL

http://www.metacritic.com/game/pc

isep instituto superior de engenharia do porto

LOWEST ERROR SO FAR: 100%