Text Mining

Exercise: Business Intelligence (Part 7)

Summer Term 2014

Stefan Feuerriegel

Today's Lecture

Objectives

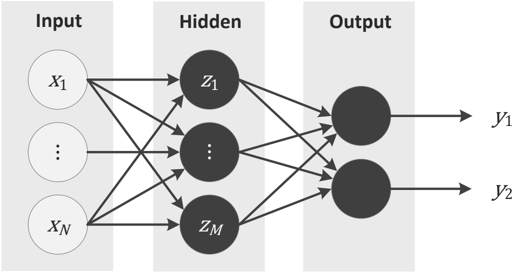
1. Being able to perform preprocessing steps for text mining
2. Learning the representation as a term-document matrix
3. Understanding how a dictionary-based sentiment analysis works

Outline

Outline

Artificial Neural Networks

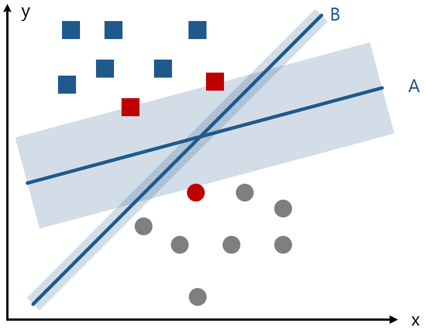
- Neurons are arranged in three (or more) layers
  - First layer: Input neurons receive the input vector $\boldsymbol{x} \in X$
  - Hidden layer(s): Connect input and output neurons
  - Final layer: Output neurons compute a response $\tilde{\boldsymbol{y}} \in Y$



- When neurons are connected as a directed graph without cycles, this is called a feed-forward ANN

Support Vector Machine (SVM)

- Which of these linear separators is optimal?
- Idea: Maximize separating margin (here: A)
  - Data points on the margin are called support vectors
  - When calculating decision boundary, only support vectors matter; other training data is ignored
  - Formulation as convex optimization problem with global solution



Predictive Performance

Confusion matrix (also named contingency table or error matrix) displays predictive performance

	Condition (as determined by Gold standard)		
	True	**False**	
Positive Outcome	True Positive (TP)	False Positive (FP) $\rightarrow$ Type I Error $\rightarrow$ False Alarm	Precision or Positive Predictive Value $= \frac{TP}{TP+FP}$
Negative Outcome	False Negative (FN) $\rightarrow$ Type II Error / Miss	True Negative (TN)	
	Sensitivity† = TP Rate $= \frac{TP}{TP+FN}$	Specificity = TN Rate $= \frac{TN}{FP+TN}$	**Accuracy** $= \frac{TP+TN}{\text{Total}}$

† Equivalent with hit rate and recall

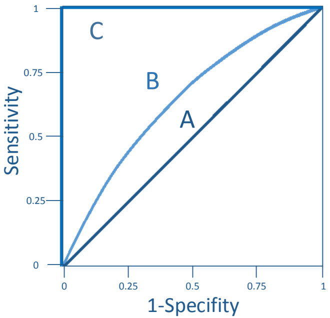
Receiver Operating Characteristic (ROC)

ROC illustrates trade-off between sensitivity and specificity

Interpretation:

- Curve A is random guessing (50 % correct guesses)
- Curve from model B performs better than A, but worse than C
- Curve C from perfect prediction

Area south-east of curve is named area under the curve and should be maximized



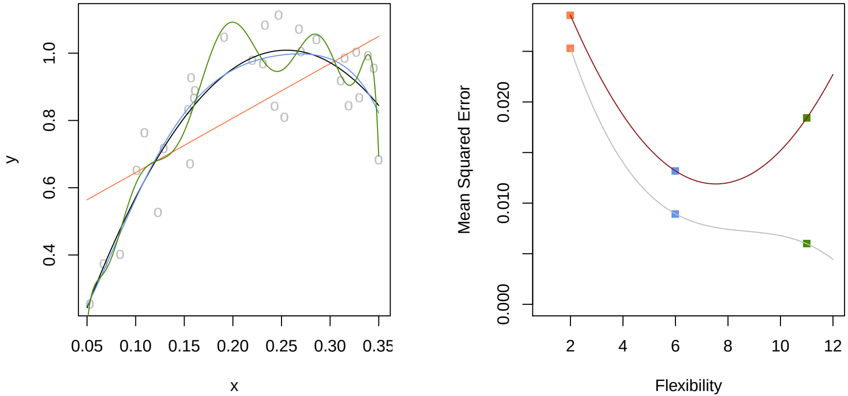
Predictive vs. Explanatory Power

Significant difference between predicting and explaining:

1. Empirical Models for Prediction
   - Empirical predictive models (e. g. statistical models, methods from data mining) designed to predict new/future observations
   - Predictive Analytics describes the evaluation of the predictive power, such as accuracy or precision
2. Empirical Models for Explanation
   - Any type of statistical model used for testing causal hypothesis
   - Use methods for evaluating the explanatory power, such as statistical tests or measures like R^2

Overfitting

- When learning algorithm is performed for too long, the learner may adjust to very specific random features not related to the target function
- Overfitting: Performance on training data (in gray) still increases, while the performance on unseen data (in red) becomes worse



Outline

Text Mining

- Text mining seeks patterns in textual content, i. e. unstructured data
- Idea: Impose (mathematical) structure first, then analyze it
- Examples:
  - Summarization
  - Categorization
  - Information extraction
  - Sentiment analysis
- Load necessary library `tm` in R to do text mining

```r
library(tm)
```

Outline

Creating the Corpus

- Collection of textual materials are called corpus
- Sources can vary from XML to text files, as well as data frames
- `Corpus(...)` creates data representation from chosen source
- Frequently annotated by additional metadata (e. g. time stamps)
- `inspect(corpus)` displays the structure of a corpus

Example:

- Access sample corpus consisting of Reuters crude oil news

```
reut21578 <- system.file("texts", "crude", package="tm")
reuters <- Corpus(DirSource(reut21578),
               readerControl=list(reader=readReut21578XML))
```

Outline

Corpus Transformation

- Additional operations necessary to transform unstructured text into a mathematical representation
- Perform transformations via `tm_map(corpus, trafo)`

  1. Remove all non-text tokens
  2. Make all letters lower case
  3. Remove redundant, non-discriminating tokens (numbers & stopwords)
  4. Reduce all inflected word forms to common base, i. e. the stem

- Example:

  "Details are given in Section 2." $\rightarrow$ "detail are giv in sect"

Example: Removing HTML/XML Tags

```r
# Corpus contains documents in XML format; remove the XML tags
if (packageVersion("tm")$minor <= 5) {
    reuters <- tm_map(reuters, as.PlainTextDocument)
} else {
    reuters <- tm_map(reuters, PlainTextDocument)
}
inspect(reuters[1])
```

```
## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $`reut-00001.xml`
## DIAMOND SHAMROCK (DIA) CUTS CRUDE PRICES
## NEW YORK, FEB 26 -
## Diamond Shamrock Corp said that
## effective today it had cut its contract prices for crude oil by
## 1.50 dlrs a barrel.
##     The reduction brings its posted price for West Texas
## Intermediate to 16.00 dlrs a barrel, the copany said.
##     "The price reduction today was made in the light of falling
## oil product prices and a weak crude oil market," a company
## spokeswoman said.
##     Diamond is the latest in a line of U.S. oil companies that
## have cut its contract, or posted, prices over the last two days
## citing weak oil markets.
##  Reuter
```

Example: Stripping Whitespaces

```
reuters <- tm_map(reuters, stripWhitespace)
inspect(reuters[1])


## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $`reut-00001.xml`
## DIAMOND SHAMROCK (DIA) CUTS CRUDE PRICES
## NEW YORK, FEB 26 -
## Diamond Shamrock Corp said that effective today it had cut its contract prices for crude oil by
```

Example: Removing punctuations

```
reuters <- tm_map(reuters, removePunctuation)
inspect(reuters[1])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $`reut-00001.xml`
## DIAMOND SHAMROCK DIA CUTS CRUDE PRICES
## NEW YORK FEB 26
## Diamond Shamrock Corp said that effective today it had cut its contract prices for crude oil by
```

Example: Converting to Lower Case

```
reuters <- tm_map(reuters, tolower)
inspect(reuters[1])


## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $`reut-00001.xml`
## diamond shamrock dia cuts crude prices
## new york feb 26
## diamond shamrock corp said that effective today it had cut its contract prices for crude oil by
```

Example: Removing Numbers

```
reuters <- tm_map(reuters, removeNumbers)
inspect(reuters[1])


## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $`reut-00001.xml`
## diamond shamrock dia cuts crude prices
## new york feb
## diamond shamrock corp said that effective today it had cut its contract prices for crude oil by
```

Stopwords

- Stopwords are short function words
- Occur frequently but no deep meaning
- Removal of stopwords in order to concentrate on more important words (that are unique/specific for the text)
- Examples: the, is, at, which, and on
- Common approach is to use predefined list of stopwords
- Get such a built-in list via `stopwords(language)`

```r
sw <- stopwords("english")
length(sw)

## [1] 174

head(sw)

## [1] "i"       "me"      "my"      "myself"  "we"      "our"
```

Example: Removing Stopwords

```
reuters <- tm_map(reuters, removeWords, stopwords("english"))
inspect(reuters[1])
```

```
## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $`reut-00001.xml`
## diamond shamrock dia cuts crude prices
## new york feb
## diamond shamrock corp said  effective today   cut  contract prices  crude oil   dlrs  barrel
```

Stemming

- Stemming is the process of reducing inflected (or sometimes derived) words to their stem, base or root form
- Depending on the algorithm, the stem is not a valid root form, but a shorted form without an ending
- Aims to group words with (possibly) the same meaning
- Examples:
  - fishing, fished, fish, fisher $\rightarrow$ fish
  - argue, argued, argues, arguing, argus $\rightarrow$ argu
  - argument and arguments $\rightarrow$ argument

Example: Stemming

```
reuters <- tm_map(reuters, stemDocument, language = "english")
inspect(reuters[1])
```

```
## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $`reut-00001.xml`
## diamond shamrock dia cut crude price
## new york feb
## diamond shamrock corp said  effect today   cut  contract price  crude oil   dlrs  barrel  redu
```

Summary: Corpus Transformations

- Perform transformations via `tm_map(corpus, trafo)`

R Function	Transformation Rule
`PlainTextDocument`	Remove HTML/XML tags
`stripWhitespace`	Eliminate unnecessary spaces, e. g. line breaks
`removePunctuation`	Remove punctuation
`tolower`	Convert to lower case letters
`removeNumbers`	Remove all numbers
`removeWords`	Remove stopwords given by additional parameter
`stemDocument`	Reduce inflected words to stem

$\rightarrow$ Results can be represented as a term-document matrix for further evaluation

Outline

2 Text Mining
- Creating the Corpus
- Transforming the Corpus
- Term-Document Matrix

Term-Document Matrix

- Term-document matrix is a mathematical matrix that describes the frequency of terms occurring in documents
- Example:
  - D_1 = "I like programming"
  - D_2 = "I hate hate programming"
  - Term-document matrix given by

	D_1	D_2
I	1	1
like	1	0
hate	0	2
programming	1	1

- Term-document matrix is input to further machine learning procedures, such as clustering, classification or prediction

Term-Document Matrix

- Create matrix via `TermDocumentMatrix(corpus)` from corpus

```
tdm <- TermDocumentMatrix(reuters)
inspect(tdm[200:205, 1:5])

## A term-document matrix (6 terms, 5 documents)
##
## Non-/sparse entries: 4/26
## Sparsity           : 87%
## Maximal term length: 10
## Weighting          : term frequency (tf)
##
##             Docs
## Terms         127 144 191 194 211
##   dhabi         0   0   0   0   0
##   dia           1   0   0   0   0
##   diamond       3   0   0   0   0
##   differenti    0   1   0   0   0
##   difficulti    0   0   0   0   0
##   dillard       0   1   0   0   0
```

Term-Document Matrix

- Use `findFreqTerms(tdm, n)` to find terms that occur at least `n` times

```
# Retrieve words that occur at least 10 times
findFreqTerms(tdm, 10)

##  [1] "accord"    "analyst"   "arabia"    "barrel"    "bpd"
##  [6] "crude"     "dlrs"      "futur"     "govern"    "group"
## [11] "increas"   "industri"  "kuwait"    "last"      "march"
## [16] "market"    "meet"      "minist"    "mln"       "month"
## [21] "new"       "offici"    "oil"       "one"       "opec"
## [26] "output"    "pct"       "petroleum" "post"      "price"
## [31] "produc"    "product"   "quota"     "report"    "reserv"
## [36] "reuter"    "said"      "saudi"     "say"       "sheikh"
## [41] "studi"     "will"      "world"     "year"
```

Text Mining Operations

- Associations are terms that frequently occur together in documents
- Measured by correlation between rows in term-document matrix
- `findAssocs(tdm, term, p)` finds associations with a correlation of at least `p` for a term

```
# Find associations for the term 'opec' with a correlation of at least 0.8
findAssocs(tdm, "opec", 0.8)

##    meet analyst    name     oil    want   emerg   buyer    said     tri
##    0.90    0.86    0.84    0.84    0.84    0.82    0.81    0.81    0.81
```

Sparsity of Term-Document Matrix

- Problem: Term-document matrices get very big, with many entries at zero
- Removal of these so-called sparse entries by deleting words that occur in less than `p` (in %) of all documents
  → `removeSparseTerms(tdm, p)`

```
# Removes words that occur in less than 40% of documents
tdm.rm.sparse <- removeSparseTerms(tdm, 0.4)
inspect(tdm.rm.sparse[, 1:5])

## A term-document matrix (6 terms, 5 documents)
##
## Non-/sparse entries: 23/7
## Sparsity           : 23%
## Maximal term length: 6
## Weighting          : term frequency (tf)
##
##         Docs
## Terms    127 144 191 194 211
##   barrel   2   0   1   1   0
##   march    0   1   0   0   0
##   oil      5  12   2   1   2
##   price    6   7   2   2   0
##   reuter   1   3   1   1   1
##   said     3  11   1   1   3
```

Analyzing a Dictionary of Terms

- Study only a subset of words of interest, specified by `dictionary = ...`

```
# select relevant terms of interest
d <- c("price", "crude", "oil")
# term-document matrix is created only for those entries
tdm.small <- TermDocumentMatrix(reuters, list(dictionary = d))
inspect(tdm.small[, 1:5])

## A term-document matrix (3 terms, 5 documents)
##
## Non-/sparse entries: 12/3
## Sparsity           : 20%
## Maximal term length: 5
## Weighting          : term frequency (tf)
##
##        Docs
## Terms   127 144 191 194 211
##   crude   3   0   3   4   0
##   oil     5  12   2   1   2
##   price   6   7   2   2   0
```

Summary: Term-Document Matrix

- Create term-document matrix from corpus via `TermDocumentMatrix(corpus)`

R Function	Inspection
`findFreqTerms(tdm, n)`	Terms occurring at least `n` times
`findAssocs(tdm, term, p)`	Terms with a correlation of at least `p`
`removeSparseTerms(tdm, p)`	Delete sparse terms with many zeros
`dictionary = ...`	Select a subset of words

$\rightarrow$ Term-document matrix is input to machine learning procedures, such as clustering, classification or prediction

Document Clustering by *k*-Means

Example: Term-document matrix can be used to cluster documents according to content using *k*-means

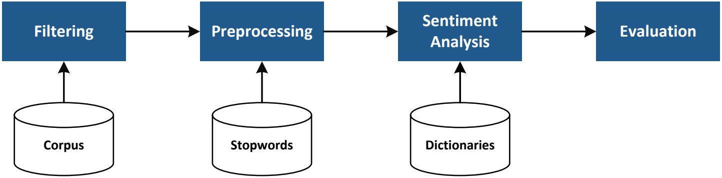
```
kmeans(t(tdm.small), 2)
```

```
## K-means clustering with 2 clusters of sizes 15, 5
##
## Cluster means:
##   crude   oil price
## 1   1.0 3.533   2.0
## 2   2.2 7.600   7.2
##
## Clustering vector:
## 127 144 191 194 211 236 237 242 246 248 273 349 352 353 368 489 502 543
##   2   2   1   1   1   2   1   1   1   2   2   1   1   1   1   1   1   1
## 704 708
##   1   1
##
## Within cluster sum of squares by cluster:
## [1] 87.73 74.80
##  (between_SS / total_SS =  50.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"
```

Outline

From News to Sentiment

- Methods that use the textual representation of documents to measure the positivity and negativity of the content are referred to as opinion mining or sentiment analysis
- Flow diagram



Sentiment Analysis

- Frequent approach utilizes dictionaries containing words labeled as positive or negative
- Let W_{pos} denote the number of positive words, W_{neg} the negative and W_{tot} the total number of words
- So-called Net-Optimism sentiment $S_{\text{NO}} \in [-1, +1]$ is given by

$$S_{\text{NO}} = \frac{W_{\text{pos}} - W_{\text{neg}}}{W_{\text{tot}}}$$

- Gives normalized ratio between positive and negative terms

Example

During the first nine months of 2008 KRONES remained on course for growth, despite the cyclical downturn. On a like-for-like basis, sales rose by 12.5 % to reach Euro 1,765.9 m. During the period under review, the company benefited from the increasing number of clients looking for all-inclusive jobs. Another growth driver during the year's first three quarters was the group's Plastics Technology Division. KRONES is the world's leading vendor of machines and ...

- Positive words marked in blue
- Negative words marked in red

$\rightarrow S_{\text{NO}} = \frac{7-1}{68} = 0.088$

Sentiment Analysis in R

- Read dictionaries with positive/negative words into data frame
- Create corresponding term-document matrices

```
pos <- as.data.frame(read.csv("positivity.txt",
                               header=FALSE))
tdm.pos <- TermDocumentMatrix(reuters,
                               list(dictionary = t(pos)))

neg <- as.data.frame(read.csv("negativity.txt",
                               header=FALSE))
tdm.neg <- TermDocumentMatrix(reuters,
                               list(dictionary = t(neg)))
```

Sentiment Analysis in R

- Calculate Net-Optimism sentiment for each document

```
# Initialize empty vector to store results
sentiment <- numeric(length(reuters))

# Iterate over all documents
for (i in 1:length(reuters)) {
    # Calculate Net-Optimism sentiment
    sentiment[i] <- (sum(tdm.pos[, i]) - sum(tdm.neg[, i]))/sum(tdm[, i])
}

# Output results
sentiment


##  [1] -0.045455  0.007273 -0.042553  0.000000  0.000000 -0.011236  0.014815
##  [8]  0.021053 -0.005208 -0.018265 -0.027778  0.000000 -0.012987 -0.028986
## [15]  0.013889  0.000000  0.008547 -0.032258  0.005291  0.024390
```

$\rightarrow$ Sentiment scores are input to data analysis (e. g. regression) or prediction (e. g. Support Vector Machine)