Text Mining with R – Twitter Data Analysis[1]

Yanchang Zhao
http://www.RDataMining.com

R and Data Mining Workshop
for the Master of Business Analytics course, Deakin University, Melbourne

28 May 2015

[1] Presented at AusDM 2014 (QUT, Brisbane) in Nov 2014 and at UJAT (Mexico) in Sept 2014

Outline

Text Mining

- unstructured text data
- text categorization
- text clustering
- entity extraction
- sentiment analysis
- document summarization
- ...

Text mining of Twitter data with R [2]

1. extract data from Twitter
2. clean extracted data and build a document-term matrix
3. find frequent words and associations
4. create a word cloud to visualize important words
5. text clustering
6. topic modelling

[2] Chapter 10: Text Mining, *R and Data Mining: Examples and Case Studies*. http://www.rdatamining.com/docs/RDataMining.pdf

Outline

Retrieve Tweets

Retrieve recent tweets by @RDataMining

```
## Option 1: retrieve tweets from Twitter
library(twitteR)
tweets <- userTimeline("RDataMining", n = 3200)
```

```
## Option 2: download @RDataMining tweets from RDataMining.com
url <- "http://www.rdatamining.com/data/rdmTweets-201306.RData"
download.file(url, destfile = "./data/rdmTweets-201306.RData")
```

```
## load tweets into R
load(file = "./data/rdmTweets-201306.RData")
```

```r
(n.tweet <- length(tweets))

## [1] 320

# convert tweets to a data frame
tweets.df <- twListToDF(tweets)
dim(tweets.df)

## [1] 320  14

for (i in c(1:2, 320)) {
    cat(paste0("[", i, "] "))
    writeLines(strwrap(tweets.df$text[i], 60))
}

## [1] Examples on calling Java code from R http://t.co/Yg1Aiv...
## [2] Simulating Map-Reduce in R for Big Data Analysis Using
## Flights Data http://t.co/uIAh6PgvQv via @rbloggers
## [320] An R Reference Card for Data Mining is now available on
## CRAN. It lists many useful R functions and packages for
## data mining applications.
```

Outline

```r
library(tm)
# build a corpus, and specify the source to be character vectors
myCorpus <- Corpus(VectorSource(tweets.df$text))

# convert to lower case
# tm v0.6
myCorpus <- tm_map(myCorpus, content_transformer(tolower))
# tm v0.5-10
# myCorpus <- tm_map(myCorpus, tolower)

# remove URLs
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
# tm v0.6
myCorpus <- tm_map(myCorpus, content_transformer(removeURL))
# tm v0.5-10
# myCorpus <- tm_map(myCorpus, removeURL)
```

```r
# remove anything other than English letters or space
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeNumPunct))

# remove punctuation
# myCorpus <- tm_map(myCorpus, removePunctuation)
# remove numbers
# myCorpus <- tm_map(myCorpus, removeNumbers)

# add two extra stop words: "available" and "via"
myStopwords <- c(stopwords('english'), "available", "via")
# remove "r" and "big" from stopwords
myStopwords <- setdiff(myStopwords, c("r", "big"))
# remove stopwords from corpus
myCorpus <- tm_map(myCorpus, removeWords, myStopwords)

# remove extra whitespace
myCorpus <- tm_map(myCorpus, stripWhitespace)
```

```r
# keep a copy of corpus to use later as a dictionary for stem completio
myCorpusCopy <- myCorpus
# stem words
myCorpus <- tm_map(myCorpus, stemDocument)
```

```r
# inspect the first 5 documents (tweets)
# inspect(myCorpus[1:5])
# The code below is used for to make text fit for paper width
for (i in c(1:2, 320)) {
  cat(paste0("[", i, "] "))
  writeLines(strwrap(as.character(myCorpus[[i]]), 60))
}

## [1] exampl call java code r
## [2] simul mapreduc r big data analysi use flight data rblogger
## [320] r refer card data mine now cran list mani use r function
## packag data mine applic
```

```r
# tm v0.5-10
# myCorpus <- tm_map(myCorpus, stemCompletion)
# tm v0.6
stemCompletion2 <- function(x, dictionary) {
  x <- unlist(strsplit(as.character(x), " "))
  # Unexpectedly, stemCompletion completes an empty string to
  # a word in dictionary. Remove empty string to avoid above issue.
  x <- x[x != ""]
  x <- stemCompletion(x, dictionary=dictionary)
  x <- paste(x, sep="", collapse=" ")
  PlainTextDocument(stripWhitespace(x))
}
myCorpus <- lapply(myCorpus, stemCompletion2, dictionary=myCorpusCopy)
myCorpus <- Corpus(VectorSource(myCorpus))
```

```
## [1] example call java code r
## [2] simulating mapreduce r big data analysis use flights data
## rbloggers
## [320] r reference card data miner now cran list use r function
## package data miner application
```

[3]

[3] http://stackoverflow.com/questions/25206049/stemcompletion-is-not-working

```r
# count frequency of "mining"
miningCases <- lapply(myCorpusCopy,
      function(x) { grep(as.character(x), pattern = "\\<mining")} )
sum(unlist(miningCases))

## [1] 82

# count frequency of "miner"
minerCases <- lapply(myCorpusCopy,
     function(x) {grep(as.character(x), pattern = "\\<miner")} )
sum(unlist(minerCases))

## [1] 5

# replace "miner" with "mining"
myCorpus <- tm_map(myCorpus, content_transformer(gsub),
                   pattern = "miner", replacement = "mining")
```

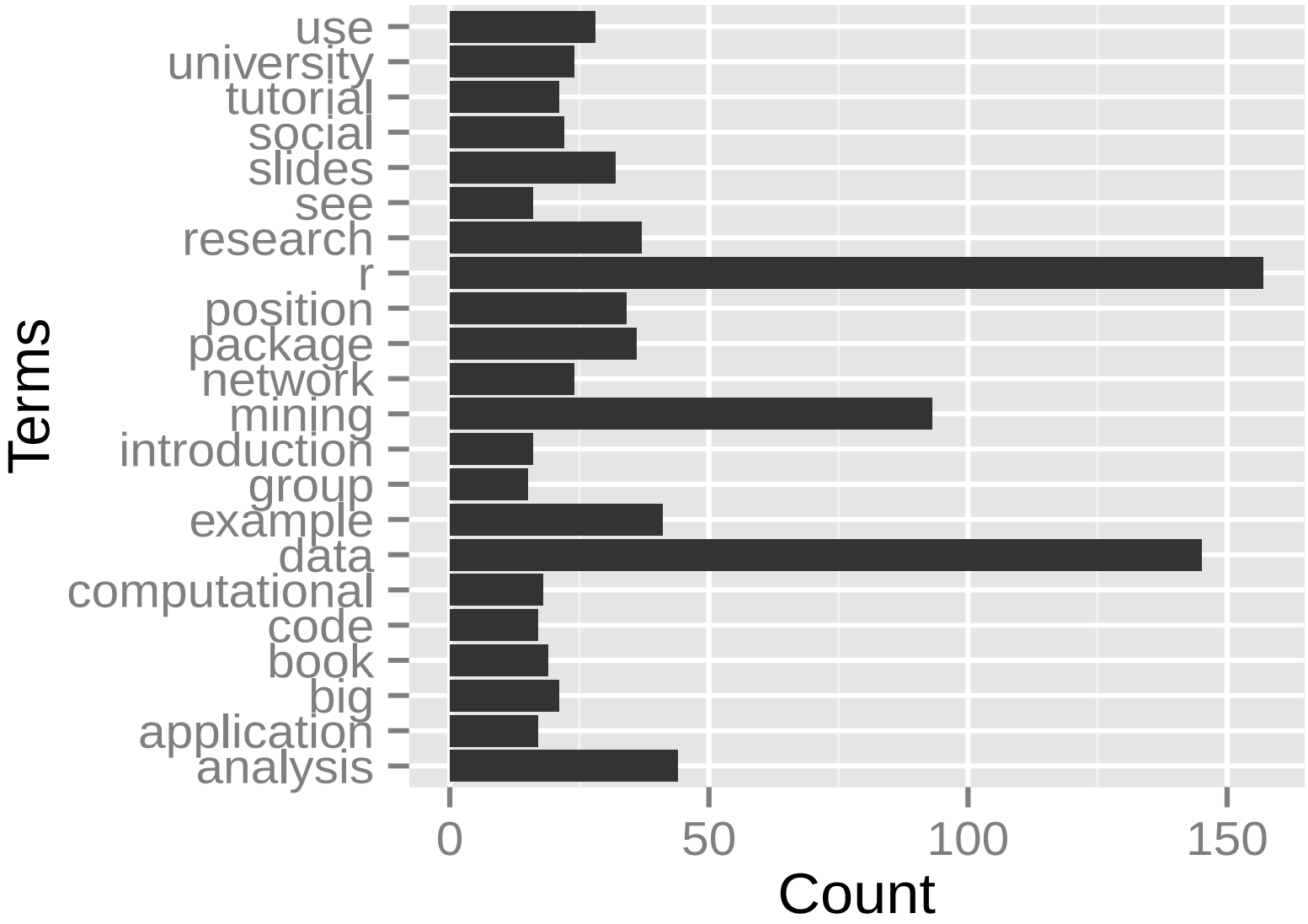
```r
tdm <- TermDocumentMatrix(myCorpus,
                          control = list(wordLengths = c(1, Inf)))
tdm

## <<TermDocumentMatrix (terms: 822, documents: 320)>>
## Non-/sparse entries: 2460/260580
## Sparsity           : 99%
## Maximal term length: 27
## Weighting          : term frequency (tf)
```

Outline

```r
idx <- which(dimnames(tdm)$Terms == "r")
inspect(tdm[idx + (0:5), 101:110])

## <<TermDocumentMatrix (terms: 6, documents: 10)>>
## Non-/sparse entries: 4/56
## Sparsity           : 93%
## Maximal term length: 12
## Weighting          : term frequency (tf)
##
##                Docs
## Terms           101 102 103 104 105 106 107 108 109 110
##   r               0   1   1   0   0   0   0   0   1   1
##   ramachandran    0   0   0   0   0   0   0   0   0   0
##   random          0   0   0   0   0   0   0   0   0   0
##   ranked          0   0   0   0   0   0   0   0   0   0
##   rann            0   0   0   0   0   0   0   0   0   0
##   rapidmining     0   0   0   0   0   0   0   0   0   0
```

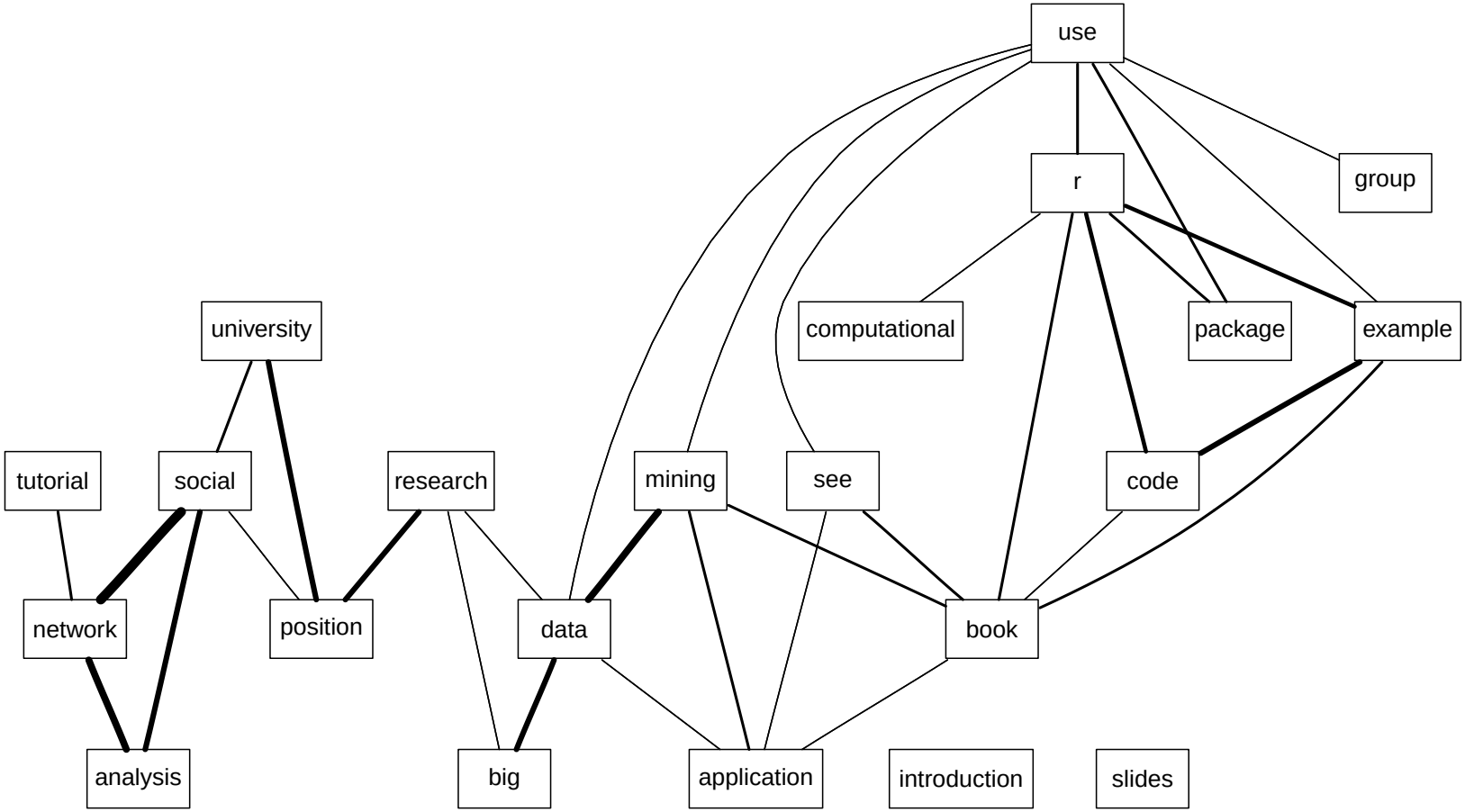
```r
# inspect frequent words
(freq.terms <- findFreqTerms(tdm, lowfreq = 15))

##  [1] "analysis"      "application"   "big"
##  [4] "book"          "code"          "computational"
##  [7] "data"          "example"       "group"
## [10] "introduction"  "mining"        "network"
## [13] "package"       "position"      "r"
## [16] "research"      "see"           "slides"
## [19] "social"        "tutorial"      "university"
## [22] "use"

term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 15)
df <- data.frame(term = names(term.freq), freq = term.freq)
```

```r
library(ggplot2)
ggplot(df, aes(x = term, y = freq)) + geom_bar(stat = "identity") +
    xlab("Terms") + ylab("Count") + coord_flip()
```



```r
# which words are associated with 'r'?
findAssocs(tdm, "r", 0.2)

##            r
## example 0.33
## code    0.29

# which words are associated with 'mining'?
findAssocs(tdm, "mining", 0.25)

##                mining
## data             0.48
## mahout           0.30
## recommendation   0.30
## sets             0.30
## supports         0.30
## frequent         0.27
## itemset          0.26
```

```r
library(graph)
library(Rgraphviz)
plot(tdm, term = freq.terms, corThreshold = 0.1, weighting = T)
```



Outline

```r
m <- as.matrix(tdm)
# calculate the frequency of words and sort it by frequency
word.freq <- sort(rowSums(m), decreasing = T)
# colors
pal <- brewer.pal(9, "BuGn")
pal <- pal[-(1:4)]
```

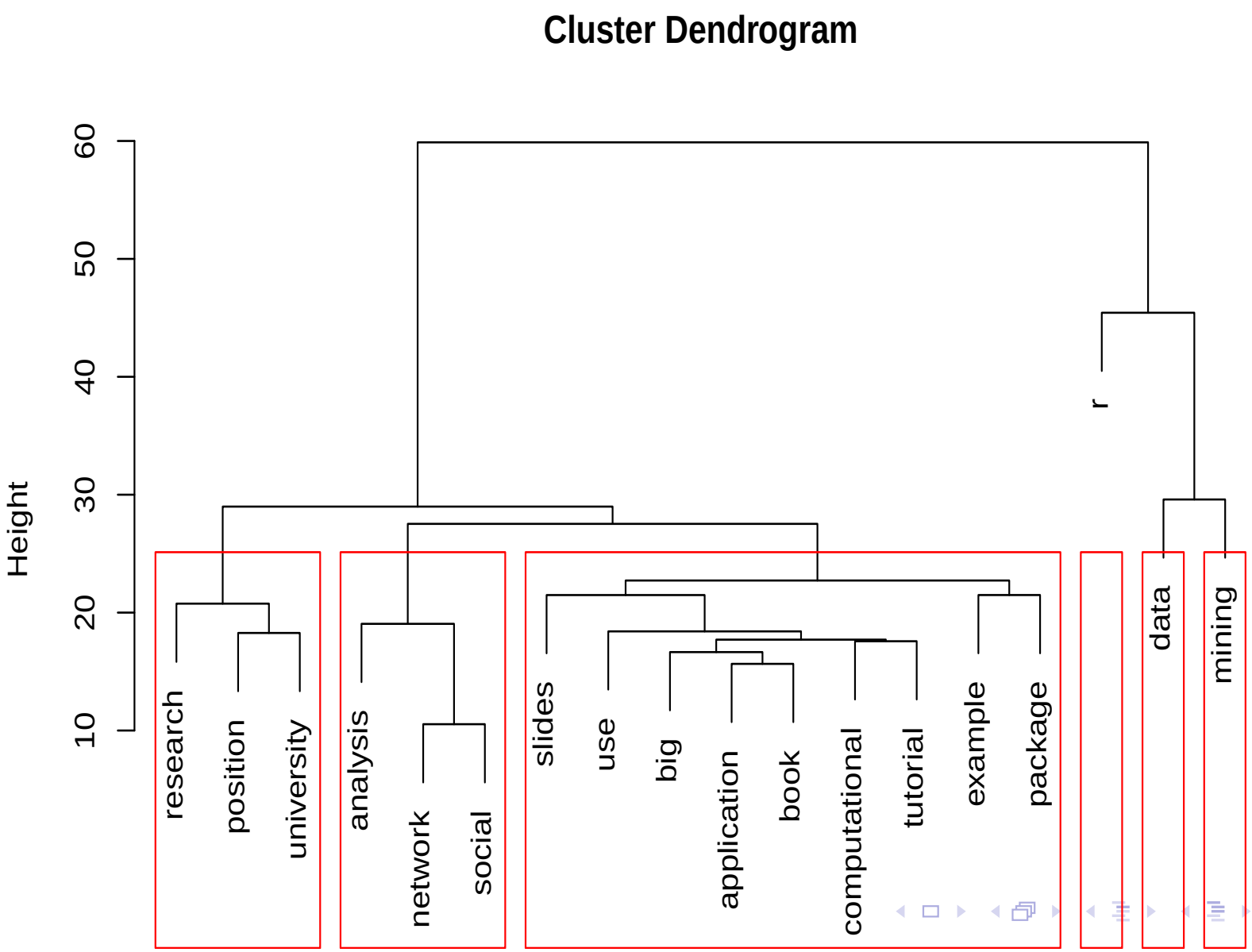
```r
# plot word cloud
library(wordcloud)
wordcloud(words = names(word.freq), freq = word.freq, min.freq = 3,
    random.order = F, colors = pal)
```


data
r
mining
package
analysis
example
research
position
slides
tutorial
university
computational
network
social
use
code
big
parallel
book
introduction
analytics
modeling
job
time
see
group
australia
associate
detection
application
postdoc
talk
snowfall
opportunity
contain
performance
dynamic
itemset
charts
search
knowledge
frequent
san
sydney
cloud
canberra
canada
visit
distributed
elsevier
studies
visualisations
thanks
initial
industrial
experience
hadoop
answers
outlier
technique
added
quick
work
plot
ibm
new
start
text
document
forecasting
week
present
due
pdf
tried
format
youtube
google
fellow
mid
amp
nd
web
interacting
software
machine
access
view
rdatamining
notes
can
project
tj
reference
statistical
chapter
southern
scientist
graph
may
sentiment
case
now
fast
create
published
csiro
edited
watson
list
large
simple
technological
engineer
short
online
cfp
recent
china
th
high
postdoctoral
free
us
guidance
april
user
tool
open
linkedin
video
topic
handling
find
vacancies
igraph
series
get
build
conference
graphical
jan
rstudio
business
top
follow
process
titled
website
rule
ausdm
media
detailed
card
map
area
learn
melbourne
dmapps
program
california
call
postdocresearch
language
database
event
science
lecture
course
also
senior
prediction
spatial
easier
center
lab
cluster
check
function
join
twitter
analyst
poll
dataset
draft
singapore
state
kdnuggets
page
submission
advanced
provided
comment
classification
cran
informatics
dec
management
wwwrdataminingcom
tweet
microsoft
track

Outline

```r
# remove sparse terms
tdm2 <- removeSparseTerms(tdm, sparse = 0.95)
m2 <- as.matrix(tdm2)
# cluster terms
distMatrix <- dist(scale(m2))
fit <- hclust(distMatrix, method = "ward")
```

```r
plot(fit)
rect.hclust(fit, k = 6)  # cut tree into 6 clusters
```



```r
m3 <- t(m2)  # transpose the matrix to cluster documents (tweets)
set.seed(122)  # set a fixed random seed
k <- 6  # number of clusters
kmeansResult <- kmeans(m3, k)
round(kmeansResult$centers, digits = 3)  # cluster centers
```

```
##   analysis application   big  book computational  data example
## 1    0.136       0.076 0.136 0.015         0.061 1.015   0.030
## 2    0.026       0.154 0.154 0.256         0.026 1.487   0.231
## 3    0.857       0.000 0.000 0.000         0.000 0.048   0.095
## 4    0.078       0.026 0.013 0.052         0.052 0.000   0.065
## 5    0.083       0.024 0.000 0.048         0.107 0.024   0.274
## 6    0.091       0.061 0.152 0.000         0.000 0.515   0.000
##   mining network package position     r research slides social
## 1  0.409   0.000   0.015    0.076 0.197    0.030  0.091  0.000
## 2  1.128   0.000   0.205    0.000 0.974    0.026  0.051  0.000
## 3  0.095   0.952   0.095    0.190 0.286    0.048  0.095  0.810
## 4  0.104   0.013   0.117    0.039 0.000    0.013  0.104  0.013
## 5  0.107   0.036   0.190    0.000 1.190    0.000  0.167  0.000
## 6  0.091   0.000   0.000    0.667 0.000    0.970  0.000  0.121
##   tutorial university   use
## 1    0.076      0.030 0.015
## 2    0.000      0.000 0.231
## 3    0.190      0.048 0.095
```

```r
for (i in 1:k) {
    cat(paste("cluster ", i, ":  ", sep = ""))
    s <- sort(kmeansResult$centers[i, ], decreasing = T)
    cat(names(s)[1:5], "\n")
    # print the tweets of every cluster
    # print(tweets[which(kmeansResult£cluster==i)])
}
```

```
## cluster 1:  data mining r analysis big
## cluster 2:  data mining r book example
## cluster 3:  network analysis social r position
## cluster 4:  package mining slides university analysis
## cluster 5:  r example package slides use
## cluster 6:  research position data university big
```

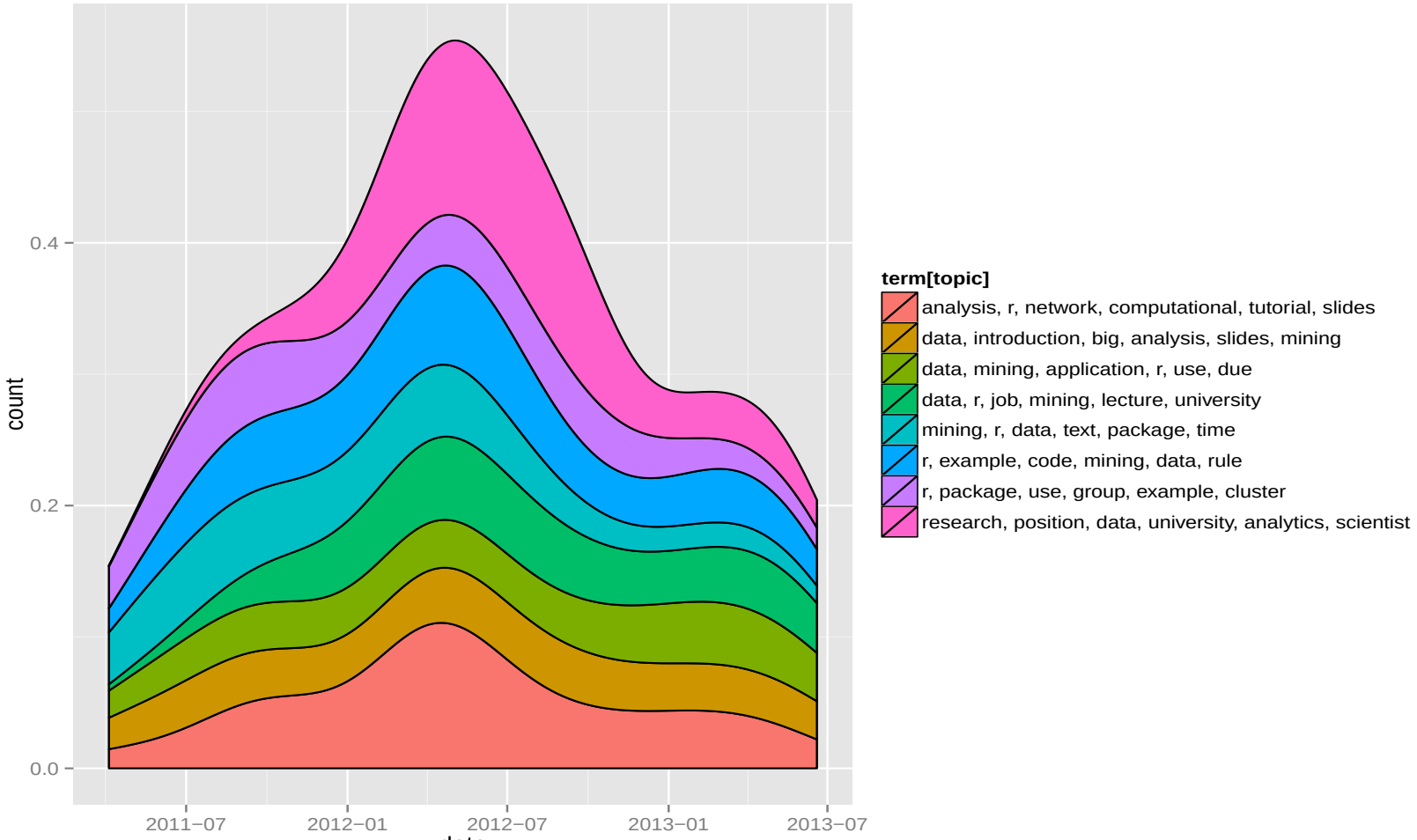
Outline

Topic Modelling

```
dtm <- as.DocumentTermMatrix(tdm)
library(topicmodels)
lda <- LDA(dtm, k = 8)  # find 8 topics
(term <- terms(lda, 6))  # first 6 terms of every topic

##      Topic 1   Topic 2        Topic 3   Topic 4   Topic 5  ...
## [1,] "r"       "data"         "mining"  "r"       "data"   ...
## [2,] "example" "introduction" "r"       "package" "mining" ...
## [3,] "code"    "big"          "data"    "use"     "applicat...
## [4,] "mining"  "analysis"     "text"    "group"   "r"      ...
## [5,] "data"    "slides"       "package" "example" "use"    ...
## [6,] "rule"    "mining"       "time"    "cluster" "due"    ...
##      Topic 6      Topic 7      Topic 8
## [1,] "data"       "research"   "analysis"
## [2,] "r"          "position"   "r"
## [3,] "job"        "data"       "network"
## [4,] "mining"     "university" "computational"
## [5,] "lecture"    "analytics"  "tutorial"
## [6,] "university" "scientist"  "slides"

term <- apply(term, MARGIN = 2, paste, collapse = ", ")
```

Topic Modelling

```
# first topic identified for every document (tweet)
topic <- topics(lda, 1)
topics <- data.frame(date=as.IDate(tweets.df$created), topic)
qplot(date, ..count.., data=topics, geom="density",
      fill=term[topic], position="stack")
```

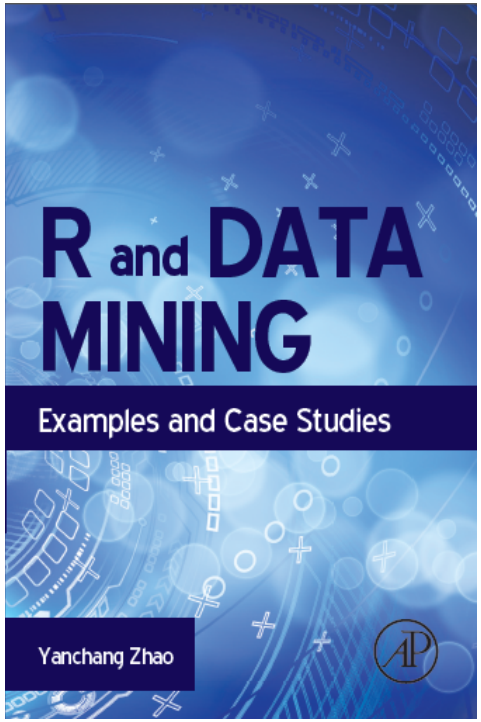


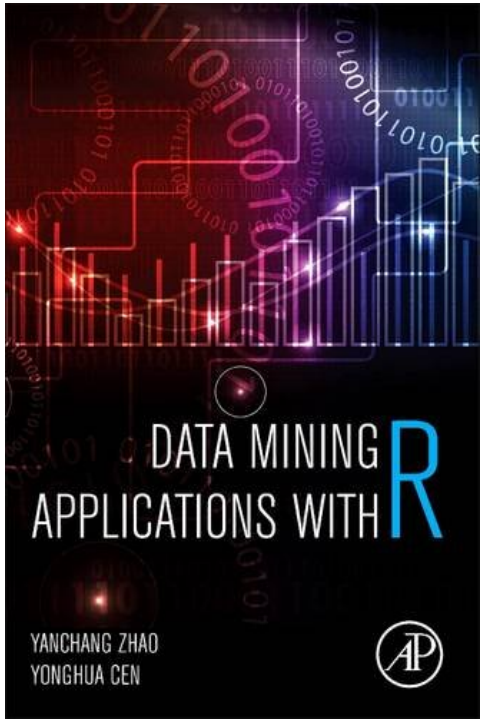
Outline

Online Resources

- Chapter 10: Text Mining, in book
  R and Data Mining: Examples and Case Studies
  `http://www.rdatamining.com/docs/RDataMining.pdf`
- R Reference Card for Data Mining
  `http://www.rdatamining.com/docs/R-refcard-data-mining.pdf`
- Free online courses and documents
  `http://www.rdatamining.com/resources/`
- RDataMining Group on LinkedIn (12,000+ members)
  `http://group.rdatamining.com`
- RDataMining on Twitter (2,000+ followers)
  `@RDataMining`

The End





Thanks!

Email: yanchang(at)rdatamining.com