R 语言环境下的文本挖掘

刘思喆

@ 上海财经大学

2012 年 11 月 3 日

目录

目录

文本挖掘的一般流程

对于文本处理过程首先要拥有分析的语料（text corpus），比如报告、出版物、网页文章等。而后根据这些语料建立半结构化的文本库（text database），生成包含词频的结构化的词条 -文档矩阵（term-document matrix）。

Figure: 文本挖掘的处理流程

文本挖掘的适用范围

解析后的结构化数据会被用于后续的分析，比如：

- 语法分析；
- 信息提取和修复；
- 文档信息汇总，比如提取相关有代表性的关键词、句子等。
- 文本分类，比如根据现有的文本分类情况，对未知文本进行归类；
- 其他

文本挖掘相关的 R 包

- XML
- tm
- topicmodels
- RWeka, lsa, RTextTools, zipfR, TextRegression, wordcloud

目录

XML 文件的解析

在 R 中对网页解析（XML、HTML 文件，或包含 XML、HTML 的字符串）有多种方法，比较成熟的方法是使用 **XML** 包。这个包能够将 XML、HTML 网页树（tree）解析成 R 结构数据。

解析函数

对标准 XML 文件的解析函数 `xmlParse` ，以及适应性更强的 `htmlTreeParse` 函数，这些函数都拥有大量的参数来适应解析需要。

解析的 XML 文件

```
<doc>
 <a xmlns:omegahat="http://www.omegahat.org">
  <b>
  <c>
    <b/>
  </c>
  </b>
  <b omegahat:status="foo">
   <r:d>
    <a status="xyz"/>
    <a/>
    <a status="1"/>
   </r:d>
  </b>
 </a>
</doc>
```

解析过程

```
doc <- xmlParse(system.file("exampleData", "tagnames.xml",
                    package = "XML"))
els <- getNodeSet(doc, "/doc//a[@status]")
sapply(els, function(el) xmlGetAttr(el, "status"))
[1] "xyz" "1"
```

XML 支持的功能远不止如此，可参考 XML 包的帮助文档

解析过程

```
doc <- xmlParse(system.file("exampleData", "tagnames.xml",
                    package = "XML"))
els <- getNodeSet(doc, "/doc//a[@status]")
sapply(els, function(el) xmlGetAttr(el, "status"))
[1] "xyz" "1"
```

XML 支持的功能远不止如此，可参考 **XML** 包的帮助文档

获得国家地震科学数据共享中心的地震数据

- 我们先观察一下它的网页
- 再看看怎么抓取
- 其实这就是一个“爬虫”

获得国家地震科学数据共享中心的地震数据

- 我们先观察一下它的网页
- 再看看怎么抓取
- 其实这就是一个“爬虫”

获得国家地震科学数据共享中心的地震数据

- 我们先观察一下它的网页
- 再看看怎么抓取
- 其实这就是一个“爬虫”

目录

中文分词工具的 R 包 rmmseg4j

rmmseg4j 调用了 Java 分词程序对语句进行分词。下面是一个例子：

```
library(rmmseg4j)
> mmseg4j('花儿为什么这样红')
[1] "花儿 为什么 这样 红"
```

这段话被分为了“花儿”，“为什么”，“这样”，“红”。这个空格分隔的结果集，后期就可以按照英文的方式进行后续处理。

由于 **rmmseg4j** 这个包已经发布 CRAN 仓库上，正常安装：

```
install.packages("rmmseg4j")
```

语料库

在 tm 中主要的管理文件的结构被称为语料库（Corpus），代表了一系列的文档集合。

动态语料库：

```
Corpus(x,
  readerControl = list(reader = x$DefaultReader, language = "en"),
  ...)
```

静态语料库 (需要 filehash 包的支持)：

```
PCorpus(x,
  readerControl = list(reader = x$DefaultReader, language = "en"),
  dbControl = list(dbName = "", dbType = "DB1"),
  ...)
```

语料库的处理

作用	函数
转化纯文本	`tm_map(reuters, as.PlainTextDocument)`
去除空白	`tm_map(reuters, stripWhitespace)`
小写变化	`tm_map(reuters, tolower)`
停止词去除	`tm_map(reuters, removeWords, stopwords("english"))`
剔除数字	`tm_map(ovid,removeNumbers)`
...	...

词条 -文档关系矩阵

词条 -文档关系矩阵是后续构建模型的基础。假设我们有两个文档分别是 text mining is funny 和 a text is a sequence of words，那么其映射到空间上对应的矩阵为：

	a	funny	is	mining	of	sequence	text	words
Doc 1	0	1	1	1	0	0	1	0
Doc 2	2	0	1	0	1	1	1	1

在 tm 包里，根据词条、文档分别作为行、列或反之，对应两种矩阵：

- TermDocumentMatrix
- DocumentTermMatrix

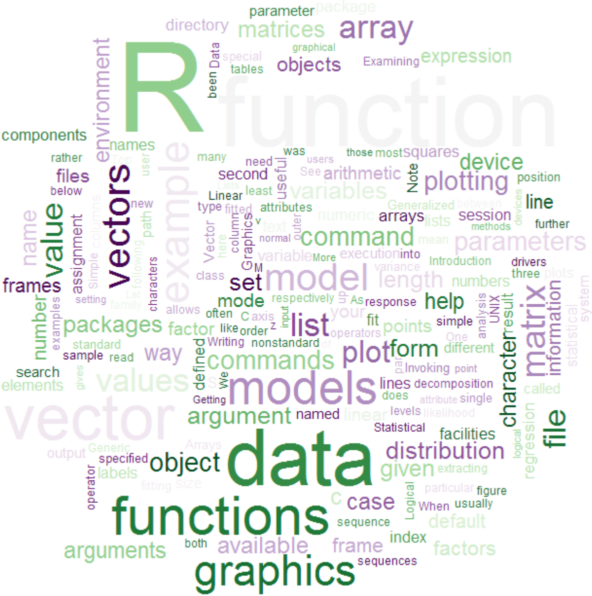
文档 -词条关系矩阵的相关操作

- `findFreqTerms(dtm, 5)`
- `findAssocs(dtm, "opec", 0.8)`
- `removeSparseTerms(dtm, 0.4)`

延伸阅读

1. tm 的扩展，包括并行计算
2. 根据条件进行文档过滤（tm_filter）
3. 元数据的管理（meta， DublinCore）
4. 字典（Dictionary）的应用

目录


parameter
package
directory
matrices
array
R
graphical
special
tables
objects
Examining
expression
Data
been
environment
function
components
names
rather
files
below
many
need
was
users
those
most
squares
device
second
See
arithmetic
Note
plotting
position
least
useful
variables
Linear
line
vectors
example
value
name
assignment
new
type
fitted
attributes
Generalized
numeric
arrays
lists
session
methods
further
path
characters
Vector
here
column
Graphics
normal
text
outer
command
mean
parameters
variable
More
execution
into
Introduction
drivers
three
plots
class
set
M
model
variance
length
numbers
frames
setting
family
allows
mode
respectively
up
As
response
help
analysis
UNIX
result
matrix
information
statistical
system
number
examples
packages
often
C
axis
input
list
your
fit
simple
factor
like
order
z
operators
points
One
standard
Writing
nonstandard
sample
read
way
plot
form
different
defined
commands
Invoking
point
search
elements
gives
values
We
models
lines
decomposition
called
character
Getting
does
attribute
single
vector
argument
named
linear
levels
likelihood
file
Statistical
facilities
output
Generic
Arrays
specified
object
data
distribution
labels
given
extracting
regression
logical
operator
fitting
size
particular
figure
functions
c
case
Logical
When
usually
sequence
default
arguments
both
available
frame
index
factors
graphics
sequences

文档识别的原理

假如我们有下表这样的矩阵，那能做分类么？

Table: 矩阵示例

Y	data	big	market	last	google	new	...
1	0	1	1	0	0	0	
1	0	0	0	0	1	1	
0	1	0	9	1	2	2	
0	0	0	2	0	1	1	
1	0	0	1	0	0	0	
0	0	1	1	0	1	0	
...							...

一般的 Classification 类的方法都可以使用在这个矩阵上，比如从最简单的 knn，稍稍高级点的 regression、decision tree，甚至 SVMs 等方法都可以使用。

主题模型简介

主题模型是专门抽象一组文档所表达“主题”的统计技术。最早的模型是 probabilistic latent semantic indexing (PLSI)，后来 Latent Dirichlet allocation (LDA，潜在狄利克雷分配模型) 模型成为了最常见的主题模型，它可以认为是 PLSI 的泛化形式。
LDA 主题模型涉及到贝叶斯理论、Dirichlet 分布、多项分布、图模型、变分推断、EM 算法、Gibbs 抽样等知识。

详细参考统计之都主站文章：

http://cos.name/2010/10/lda_topic_model/

应用实例（一）

在某个门户网站上收集了 720 篇文章，对这些文档构建 LDA 模型，参数设置为 6 类，模型提取出 6 个类别的关键词分别是美食、旅游、拍摄、购物、汽车和招聘，每个关键词对应的文档数量如下图所示：

Table: 预测种类及主题关键词

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
美食	旅游	拍摄	购物	汽车	招聘
102	140	118	110	129	121

应用实例（二）

实际上收集的这些文章分属不同频道: ”购物类”, ”旅游类”, ”美食类”, ”汽车类”, ”数码类”, ”招聘类”，共计六类。对比网站编辑的人工标记同模型标记的差异情况：

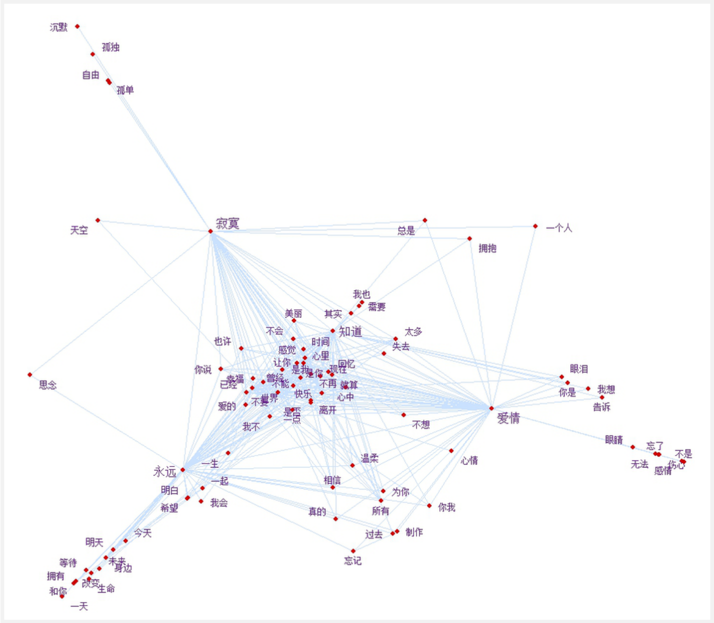
Table: 预测种类同人工判别种类的混淆矩阵

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
购物类	9	1	3	91	15	1
旅游类	0	113	0	4	0	3
美食类	90	24	0	5	1	0
汽车类	3	0	0	7	110	0
数码类	0	2	115	0	3	0
招聘类	0	0	0	3	0	117

注：在使用 topic model 过程中设置了 6 类的参数是为了和实际情况进行比对

关键词网络

- 22996 首中文流行歌曲的歌词 (844 位歌手)
- 这些歌词是怎么组织在一起的？


沉默
孤独
自由
孤单
寂寞
天空
总是
一个人
拥抱
我也
需要
美丽
其实
知道
太多
不会
时间
失去
也许
感觉
心里
让你
回忆
你说
是我
现在
幸福
曾经
已经
不能
不再
就算
眼泪
快乐
心中
你是
我想
世界
爱的
离开
告诉
爱情
不想
我不
思念
眼睛
忘了
不是
无法
感情
伤心
一生
温柔
心情
永远
一起
相信
明白
为你
希望
我会
所有
你我
真的
今天
过去
制作
明天
忘记
等待
未来
身边
拥有
改变
生命
和你
一天

爱情
永远
寂寞

- 邮件：sunbjt<at>gmail.com
- 博客: `http://www.bjt.name`
- 微博：@ 刘思喆

Jump to first slide