Instruções de como processar a Literateca em R

Diana Santos

Maio de 2020

Este mini-curso é para servir de continuação ao curso da EBraLC de 2012, e para permitir a interessados em leitura distante fazerem uso do material da Literateca, usando a linguagem R.

O número de características (número de colunas) pode, naturalmente, variar no futuro.

1 Preparativos para ter acesso ao material da Literateca

Vamos partir de duas tabelas criadas a partir do AC/DC, que se encontram, públicas, em `https://www.linguateca.pt/Gramateca/Literateca/Obras.R.txt` e `https://www.linguateca.pt/Gramateca/Literateca/Obras2.R.txt`.

Vamos ler cada uma separadamente, e juntar numa única estrutura de dados, chamada TODOS.

```
primeiro<-read.table("https://www.linguateca.pt/Gramateca/Literateca/
Obras.R.txt",header=TRUE)
segundo<-read.table("https://www.linguateca.pt/Gramateca/Literateca/
Obras2.R.txt",header=TRUE)
```

Nota: se não vir bem os caracteres acentuados, ao fazer

```
head(primeiro[,1], 10,)
```

repita o processo com os endereços ObrasUTF.R.txt e Obras2UTF.R.txt

```
todos<-merge(segundo,primeiro,by.x=c("Obra","id","tamanho","autor",
"genero"),by.y=c("Obra","id","tamanho","autor","genero"))
```

e confira o tamanho desta estrutura:

```
dim(todos)
```

O primeiro número corresponde ao número de obras (uma por linha), e o segundo ao número de características presentes. Pode ficar com uma ideia de cada uma das características fazendo

```
summary(todos)
```

Para selecionar uma característica específica, é apenas colocar esse nome após um cifrão e o nome da estrutura, por exemplo:

```
todos$autor
todos$data
todos$passiva
```

Para fazer gráficos relacionados com estes dados, é simplesmente escolher as características que nos interessam e desenhá-las.

Alguns exemplos: a variação no número de cores entre as várias obras

```
boxplot(todos$cor)
```

ou a variação entre o número relativo de cores entre as várias obras

```
boxplot(todos$cor/todos$tamanho)
```

Podemos fazer isso apenas para os textos em prosa

```
boxplot(todos[grep("^Prosa",as.character(todos$genero), perl=TRUE),]$saude)
```

ou fazer isso apenas para os romances

```
boxplot(todos[grep("romance",as.character(todos$genero), perl=TRUE),]$parentesco
```

Se se quiser tratar, por exemplo, a poesia, o teatro e a prosa separadamente, é mais fácil criar três estruturas

```
poesia<-todos[grep("^Poesia",as.character(todos$genero), perl=TRUE),]
teatro<-todos[grep("^Teatro",as.character(todos$genero), perl=TRUE),]
prosa<-todos[grep("^Prosa",as.character(todos$genero), perl=TRUE),]
```

Para saber a dimensão (número de obras e tamanho) de cada uma delas, basta fazer

```
dim(prosa)
sum(prosa$tamanho)
```

Mas também se pode ver uma característica por autor, ou por obra, ou por época. Para simplificar, vamos usar apenas a prosa, e escusamos de escrever o nome todo se fizermos

```
attach(prosa)
```

Observando o número de passivas:

```
boxplot(passiva~autor)
```

Ou o número relativo de passivas

```
boxplot(passiva/tamanho~autor)
```

Claro que se pode fazer comparações mais finas: por exemplo selecionar apenas quatro autores:

```
exp<-subset(todos,autor=="LimBar" | autor=="CoeNet" | autor=="EcaQue"
| autor=="JulDin")
exp$autor<-exp$autor[drop=TRUE]
boxplot(exp$dizer/exp$tamanho~exp$autor)
```

Para tratar de algo ao longo do tempo, é preciso explicar ao R que coisas como 1845 são números, o que se faz da seguinte maneira (o R é bom a descobrir o tipo de colunas, mas no nosso caso temos algumas obras em que a data é `desc`, e por isso ele assume que a coluna data contém (nomes de) classes).

Vamos obter o caso das obras com data

```
todos_com_data<-todos[todos$data!="desc",]
todos_com_data$data<-todos_com_data$data[drop=TRUE]
todos_com_data$data<-as.numeric(as.character(todos_com_data$data))
```

Para confirmar que o R já trata o campo data como números, basta pedir um resumo da data:

```
summary(todos_com_data$data)
```

Agora podemos criar várias novas colunas referentes ao tempo, por exemplo década, século, etc., assim como podemos selecionar obras por períodos de tempo. Tomemos o período COST como exemplo (de 1840 a 1919), e criemos fatias de 20 anos:

```
COST<-todos_com_data[todos_com_data$data<1920 & todos_com_data$data> 1839,]
dim(COST)
decada20<-function(x) {
trunc((x-1840)/20)+1}
COST$decada20<-decada20(COST$data)
barplot(table(COST$decada20),names.arg=c("1840-","1860-","1880-","1900-"))
```

Se quiséssemos restringir apenas a romances, e como se distribuem por décadas, faríamos:

```
COST<-COST[grep("romance",as.character(COST$genero), perl=TRUE),]
dim(COST)
decada<-function(x) {
trunc((x-1840)/10)+1}
COST$decada<-decada(COST$data)
barplot(table(COST$decada),names.arg=c("1840-","1850-","1860-",
"1870-","1880-","1890-","1900-","1910-"))
```

Outro tipo de pergunta seria por exemplo a distribuição de menções a membros da família:

```
boxplot(COST$parentesco~COST$decada, names=c("1840-","1850-","1860-",
"1870-","1880-","1> 0-","1900-", "1910-"))
```

2 Métodos de exploração e de agrupamento

Estes métodos tentam encontrar estrutura e semelhanças ou diferenças entre vários elementos.

2.1 Análise de correspondências

Para usar o programa de análise de correspondências, é preciso instalar algumas bibliotecas. Se o comando

```
library(languageR)
```

se queixar de que não conhece esta biblioteca, é preciso instalar o pacote

```
install.packages("languageR")
```

e voltar a executar o comando anterior.

Para fazer uma análise de correspondências, basta apenas indicar ao R quais as colunas que devem ser tomadas em conta (e as colunas só podem ser numéricas).

```
todos.ca<-corres.fnc(todos[,-c(1,2,4,5,133,134,135,136)])
```

Se tudo correr bem, pode-se imprimir o resultado

```
plot(obras.ca,rlabels=todos$autor,extreme=0.1)
```

ou com base noutras colunas com classificação

```
plot(obras.ca,rlabels=todos$Obra,extreme=0.1)
plot(obras.ca,rlabels=todos$genero,extreme=0.1)
```

2.2 Análise de componentes principais

Esta análise é para valores reais (não contagens), por isso temos de usar os valores relativos (quantas passivas por número de palavras, quantos amarelos por número de palavras). Para isso criamos uma nova estrutura de dados com todos os valores divididos pelo tamanho, a que chamámos `todosrel`.

```
todosrel<-todos[,-c(1,2,3,4,5,133,134,135,136)]/todos$tamanho
row.names(todosrel)<-todos$Obra
todosrel.pr<-prcomp(todosrel,scale=T)
biplot(todosrel.pr,var.axes=F)
```

A opção `scale` é para evitar que as características com maior variação dominem a análise.

Os componentes principais são uma combinação linear das variáveis originais. Podemos ver o peso nos três primeiros componentes com o comando

```
todosrel.pr$rotation[,1:3]
```

2.3 Análise de fatores

É preciso pedir o número de fatores que queremos. O ficheiro `novoCoresCaract.txt` tem a cor de cada característica.

```
todosrel.fac<-factanal(todosrel, factors=3, rotation="promax")
loadings<-loadings(todosrel.fac)
cores<-read.table("http://folk.uio.no/dssantos/cursoR/novoCoresCaract.txt",
header=TRUE)
plot(loadings, type="n")
text(loadings, rownames(loadings), col=cores$cor, cex=0.8)
legend("topleft",legend=c("sintaxe","cor","roupa","saude","corpo",
"dizer","emos","familia"), col=1:8, pch=1)
```

3 Classificação

Existem outros métodos que tentam atribuir uma classificação a novos elementos, depois de terem "aprendido" com um corpo de exemplos.

Vamos experimentar com uma réplica (de momento imperfeita) do problema descrito por num artigo de Barufaldi et al. no STIL 2009.

Apresento apenas alguns exemplos de como conseguir um tal subconjunto

```
stil<-todos[todos$Obra=="Sermão_da_Primeira_Dominga_do_Advento"|todos$Obra
=="Sermão_da_Sexagésima"|todos$Obra=="Sermao_do_Espírito_Santo"|
todos$autor=="GreMat"|todos$Obra=="O_moço_louro"|...
stil$autor<-stil$autor[drop=TRUE]
stil$Obra<-stil$Obra[drop=TRUE]
stil[stil$Obra=="A_Normalista",]$escola<-"realismo"
stil[stil$Obra=="Memórias_de_um_Sargento_de_Milícias",]$escola<-"romantismo"
...
stil$escola<-stil$escola[drop=TRUE]
```

3.1 Análise de discriminantes

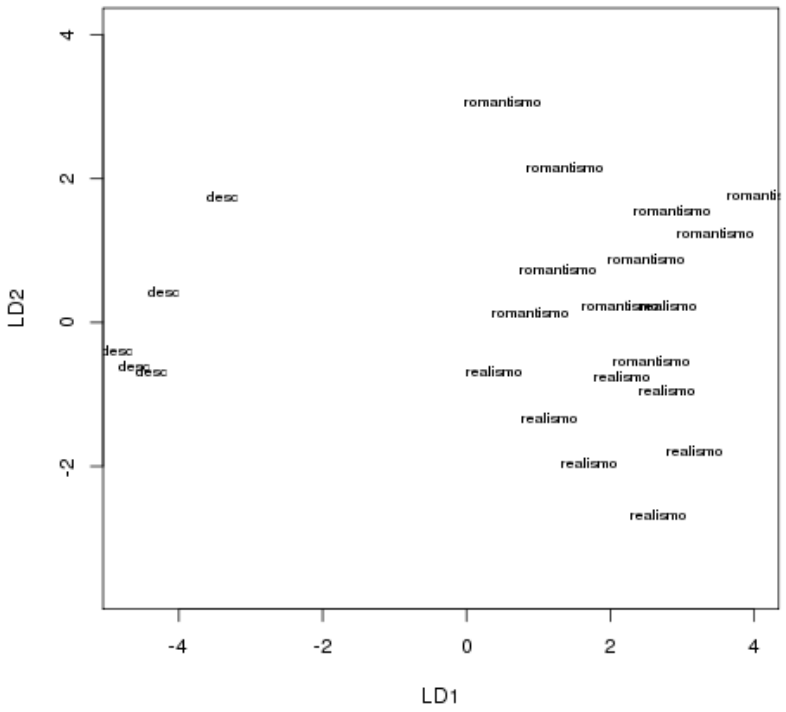
Semelhante à análise de componentes principais, exceto que os componentes, chamados discriminantes, são calculados de forma a produzir médias o mais diferentes possível para cada grupo, e cada grupo ter o mínimo de variação.

Como esta análise exige que as características sejam independentes, é costume/possível executar primeiro uma análise de componentes, e depois a análise de discriminantes com base nos componentes principais.

As colunas retiradas, em seguida, são as que correspondem aos casos em que todos os textos (do subconjunto stil) têm 0 ocorrências...

```
stilrel<-stil[,-c(1,2,3,4,5,43,77,78,90,96,105,107,110,129,133,
134,135,136)]/stil$tamanho
stilrel.pca<-prcomp(stilrel)
biplot(stilrel.pca,var.axes=F)
stilrel.x<-data.frame(stilrel.pca$x)
stilrel.x=stilrel.x[order(rownames(stilrel.x)),]
library(MASS)
stilrel.pca.lda= lda(stilrel.x[, 1:5], stil$escola)
plot(stilrel.pca.lda)
round(predict(stilrel.pca.lda,stilrel.x[,1:5])$posterior,4)
```

Se quisermos avaliar o resultado deste modelo, basta fazermos a tabulação cruzada:



Figur 1: Figura produzida pela análise de discriminantes com base nos 5 primeiros componentes principais, em que `desc` significa o barroco

```
xtabs(~stil$escola+predict(stilrel.pca.lda,stilrel.x[,1:5])$class)
```

```
stil$escola  arcadismo barroco realismo romantismo
  arcadismo          1       1        3          0
  barroco            0       5        2          0
  realismo           1       0       10          2
  romantismo         0       1        6          5
```

3.2 SVM

Outro classificador muito usado tem o nome em inglês de *support vector machines* (que eu posso talvez traduzir como "máquinas que criam vetores de apoio").

```
stilrel.svm<-svm(stilrel,stil$escola)
predict(stilrel.svm)
```

Para ver quantos casos de previsão correta ou não, pode fazer-se uma tabulação cruzada:

```
xtabs(~stil$escola + predict(stilrel.svm))
```

```
stil$escola  arcadismo barroco realismo romantismo
  arcadismo          5       0        0          0
  barroco            0       7        0          0
  realismo           0       0       13          0
  romantismo         0       0        0         12
```

4 Análise de temas (*topic modelling*)

A análise de temas é claramente a técnica estatística mais usada na literatura, e é baseada na análise de discriminantes (sobre o texto das obras).

Para fazer esta análise, temos de ter as obras divididas em lotes do mesmo tamanho, que são a entrada do programa. De qualquer maneira indico aqui todos os passos necessários para executar, tendo acesso a esses lotes.

```
library(mallet)
library(tm)
library(wordcloud)
```

Se estas bibliotecas não estiverem acessíveis, é preciso instalá-las, claro.

Depois o comando para carregar os lotes é o seguinte, seguido por outro comando que lê palavras que não devemos usar:

```
documents<-mallet.read.dir("diretoria...")
mallet.instances <- mallet.import(documents$id, documents$text,
"ARQUIVO de CASOS A NÃO USAR",TRUE)
```

O processo de criar os modelos temáticos segue depois todas estas etapas:

```
n.topics <- 100
topic.model <- MalletLDA(n.topics)
topic.model$loadDocuments(mallet.instances)
vocabulary <- topic.model$getVocabulary()
word.freqs <- mallet.word.freqs(topic.model)
topic.model$setAlphaOptimization(20, 50)
topic.model$train(200)
topic.model$maximize(10)
doc.topics <- mallet.doc.topics(topic.model, smoothed=T, normalized=T)
topic.words <- mallet.topic.words(topic.model, smoothed=T, normalized=T)
topic.docs <- t(doc.topics)
topic.docs <- topic.docs / rowSums(topic.docs)
```

Para imprimir cada tópico, podemos usar uma nuvem de palavras, como aqui:

```
for (topic in 1:n.topics) {nome<-paste0("topico",topic,".png"); png(nome);
wordcloud(words = mallet.top.words(topic.model, topic.words[topic,],
num.top.words=100)$words, freq=mallet.top.words(topic.model, topic.words[50,],
num.top.words=100)$weights, min.freq=0.001, random.order=FALSE, rot.per=0.35,
colors=brewer.pal(8, "Dark2"));dev.off()}
```

ou simplesmente escrever num arquivo as (5) palavras que constituem os tópicos

```
topics.labels <- rep("", n.topics)
for (topic in 1:n.topics) topics.labels[topic] <- paste(mallet.top.words(
topic.model, topic.words[topic,], num.top.words=5)$words, collapse=" ")
write.table(topics.labels, file="nomestopicosNVA.txt")
```

Exemplos obtidos são

```
"16" "sangue pátria nome povo liberdade glória ferro"
"26" "rei povo arma homem espada gente cidade"
"44" "padre senhor abade cónego pároco velho freguesia"
```