12

비정형데이터분석

텍스트 데이터의 통계적 분석(1)

통계·데이터과학과 장영재 교수

학습목차

1. 텍스트마이닝(Text mining)
2. 코사인유사도를 이용한 문서의 분류

01

텍스트 마이닝(Text mining)

1. 텍스트 마이닝(Text mining)

- 텍스트 데이터를 분석하여 텍스트 데이터가 내포하고 있는 정보를 발견해내는 기법을 텍스트 마이닝이라고 함
  - 텍스트 데이터의 분석에는 다양한 데이터 마이닝 기법들을 적용할 수 있음을 의미
  - 텍스트 데이터 분석에 자주 활용되는 대표적인 데이터 마이닝 기법으로는 군집분석(clustering), 분류분석(classification) 등

→ 자주 등장하는 단어들을 비교하여 문서의 저자를 찾아내는 방법의 유용성을 보인 사례도 있음(Jockers, 2014)

관리학습

분류분석
- 판별분석
- 로지스틱회귀분류
- 최근접이웃기법
- 의사결정나무
- 나이브베이즈분류
- 신경망
- 지지도벡터기계

예측분석
- 회귀분석
- 최근접이웃기법
- 신경망
- 평활법

자율학습

군집분석
- K-평균
- 계층적 군집분석
- 유한혼합모형
- 이중군집법

연관분석
- 장바구니분석
- 서열분석
- 트랜잭션데이터분석

비정형분석
- 텍스트마이닝, 사회연결망분석

<그림> 데이터마이닝 기법의 분류

군집분석

K-평균
계층적 군집분석
유한혼합모형
이중군집법

<그림> 군집분석

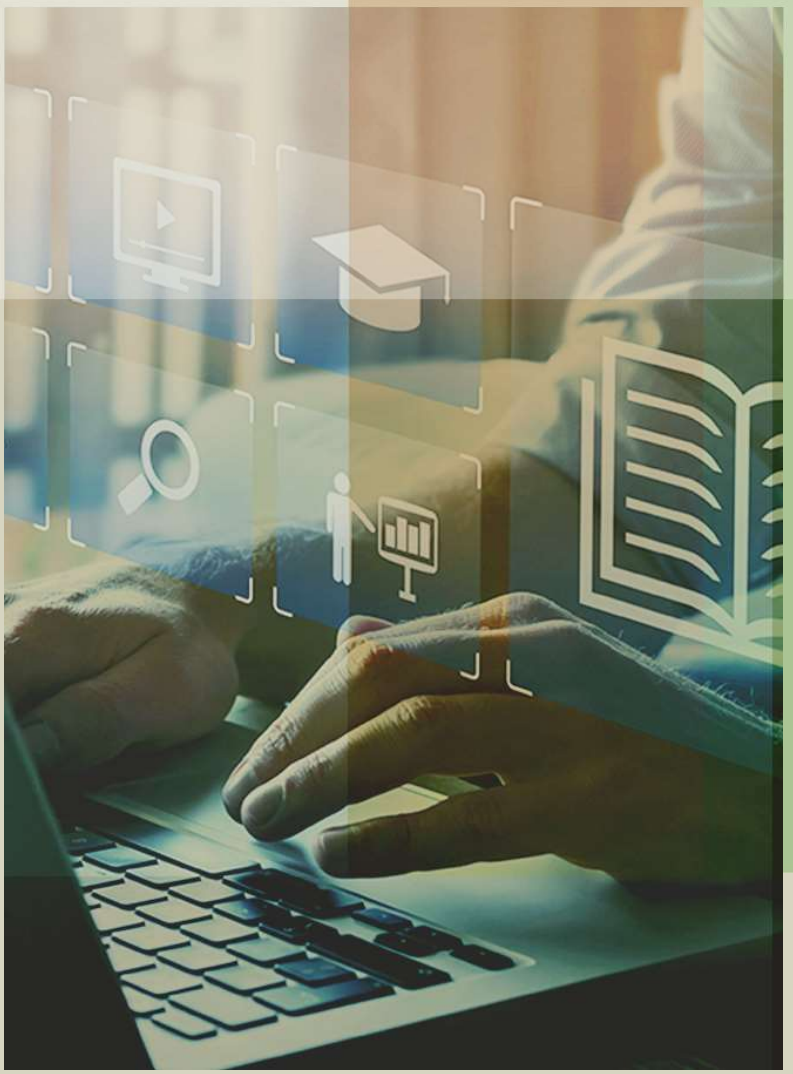
분류분석

- 판별분석
- 로지스틱회귀분류
- 최근접이웃기법
- 의사결정나무
- 나이브베이즈분류
- 신경망
- 지지도벡터기계

<그림> 분류분석

02

코사인 유사도를 이용한 문서의 분류



2. 코사인 유사도를 이용한 문서의 분류

- 문서-단어행렬로 표현된 텍스트 데이터의 코사인 유사도를 이용하여 문서들 사이의 유사성을 찾아내는 방법을 고려
  - 문서 x와 문서 y가 각각 벡터 $x = (x_1, x_2, \cdots, x_p)$ 과 $y = (y_1, y_2, \cdots, y_p)$
  - x_i와 y_i는 각각 문서 x와 y에서의 단어 w_i의 출현횟수에 해당하며 코사인 유사도는 1에 가까운 값을 가질수록 유사도가 높은 것으로 평가

$$\cos(x, y) = \frac{x \cdot y}{|x||y|} = \frac{x_1 y_1 + x_2 y_2 + \cdots + x_p y_p}{\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}\sqrt{y_1^2 + y_2^2 + \cdots + y_p^2}}$$

2. 코사인 유사도를 이용한 문서의 분류

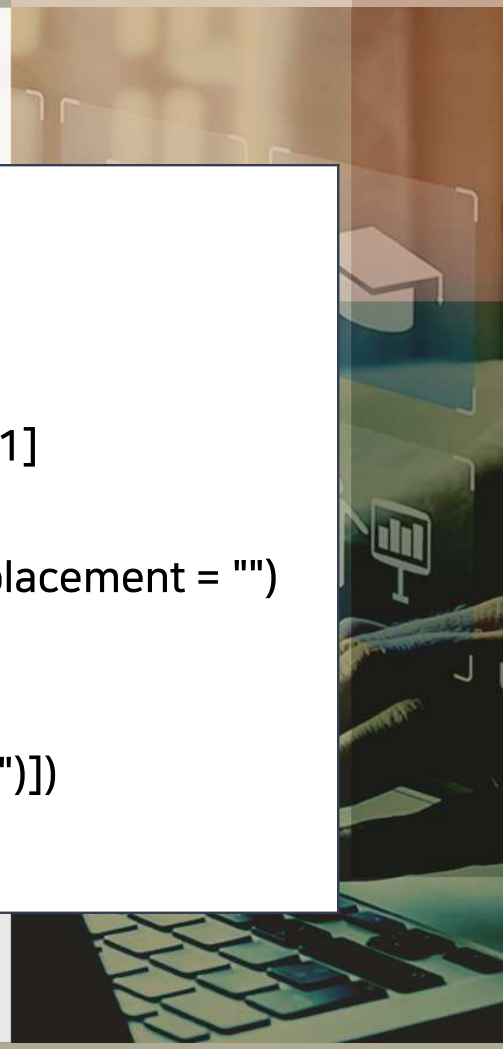
1 분석 대상 문서의 문서-단어행렬 작성

- 코사인 유사도의 개념을 적용하기 위해 문서-단어행렬 작성
  - 「로빈슨 크루소」와 「작은 아씨들」의 각 장별 비교

```
> RC <- scan("http://www.gutenberg.org/files/521/521-0.txt", what = "character", encoding = "UTF-8", sep = "₩n")
> LW <- scan("http://www.gutenberg.org/cache/epub/514/pg514.txt", what="character", encoding = "UTF-8", sep="₩n")
> RC_Chpt <- grep(RC, pattern = "CHAPTER")
> LW_Chpt <- grep(LW, pattern = "CHAPTER")
> RC_End <- grep(tolower(RC), pattern="end of the project gutenberg")-1
> LW_End <- grep(tolower(LW), pattern="end of the project gutenberg")-1
```

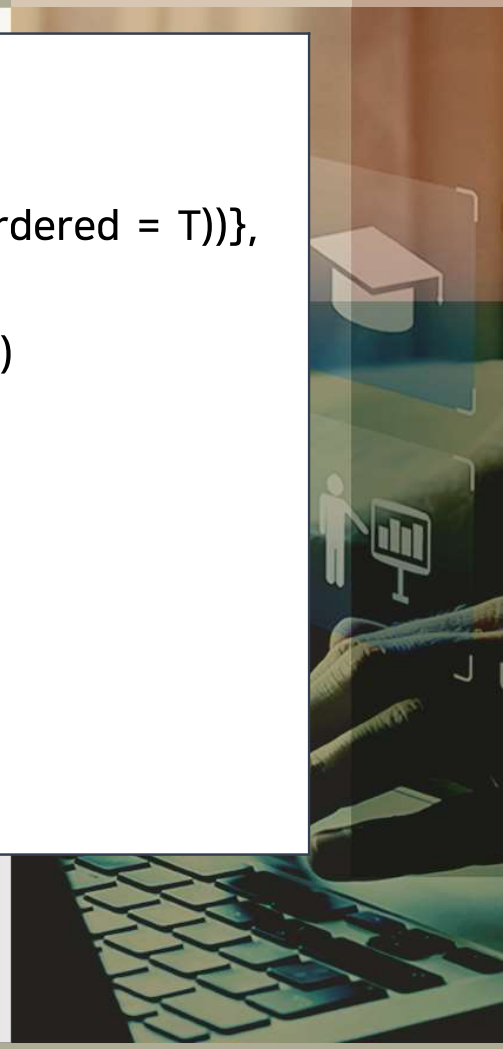
2. 코사인 유사도를 이용한 문서의 분류

```
> RC_body <- RC[(RC_Chpt[21]):RC_End]
> LW_body <- LW[(LW_Chpt[48]):LW_End]
> RCLW_body <- c(RC_body, LW_body)
> RCLW_by_Chpt <- unlist(strsplit(paste(RCLW_body, collapse=" "), "CHAPTER"))[-1]
> RCLW_by_Chpt <- gsub(x = RCLW_by_Chpt, pattern = "'s", replacement = "")
> RCLW_by_Chpt <- gsub(RCLW_by_Chpt, pattern = "([^[:alnum:][:blank:]'-])", replacement = "")
> RCLW_by_Chpt <- tolower(RCLW_by_Chpt)
> RCLW_by_Chpt <- strsplit(RCLW_by_Chpt, " ")
> RCLW_by_Chpt <- lapply(RCLW_by_Chpt, function(x) x[! x %in% c(stopwords(), "")])
> RCLW_by_Chpt <- lapply(RCLW_by_Chpt, lemmatize_strings)
```



2. 코사인 유사도를 이용한 문서의 분류

```
> RCLW_lev <- sort(unique(unlist(RCLW_by_Chpt)))
> RCLW_DTM <- lapply(RCLW_by_Chpt, FUN = function(x, lev){table(factor(x, lev, ordered = T))},
lev = RCLW_lev )
> RCLW_DTM <- matrix(unlist(RCLW_DTM), nrow = length(RCLW_DTM), byrow = TRUE)
> dim(RCLW_DTM>0)
[1]    67 9899
> sum(RCLW_DTM>0)
[1] 60252
> sum(RCLW_DTM==0)
[1] 602981
```



2. 코사인 유사도를 이용한 문서의 분류

2 문서-단어행렬을 이용한 코사인 유사도 계산

- 문서-단어행렬을 이용하여 두 소설 각 장들의 코사인 유사도 계산
  - RCLW_DTM 행렬과 전치행렬 t(RCLW_DTM)을 곱해서 구한 RCLW_DTMsqr 행렬을 이용하여 각 문서 벡터의 내적을 계산
  - RCLW_DTMsqr의 대각원소들의 벡터를 diag( ) 함수로 선택하고 이 벡터들을 곱하여 sqrt( ) 함수를 적용하여 두 문서의 길이의 곱에 해당되는 값들이 기록된 행렬을 산출

2. 코사인 유사도를 이용한 문서의 분류

2 문서-단어행렬을 이용한 코사인 유사도 계산

- 문서-단어행렬을 이용하여 두 소설 각 장들의 코사인 유사도 계산
  - 「로빈슨 크루소」는 20개 장, 「작은 아씨들」은 47개 장으로 구성되어 있으므로 일부만 살펴보기 위해 sample.int( ) 함수를 이용하여 랜덤추출
    → 「로빈슨 크루소」에서는 1, 2, 4, 7, 13장이, 「작은 아씨들」에서는 14, 18, 23, 33, 43장이 선택되었음

2. 코사인 유사도를 이용한 문서의 분류

```
> RCLW_DTMsqr <- RCLW_DTM %*% t(RCLW_DTM) # 행렬의 곱은 %*%
> RCLW_CosSim <- RCLW_DTMsqr / sqrt(diag(RCLW_DTMsqr) %*% t(diag(RCLW_DTMsqr)))
> set.seed(1)
> RCsample <- sort(sample.int(n=20, size=5))
> RCsample
[1]  1  2  4  7 13
> LWsample <- sort(sample.int(n=47, size=5))
> LWsample
[1] 14 18 23 33 43
RCLW_CosSim_smpl <- RCLW_CosSim[c(RCsample, LWsample+20), c(RCsample, LWsample+20)]

> rownames(RCLW_CosSim_smpl) <- c(paste0("Robinson",RCsample),paste0("Women",LWsample))
> colnames(RCLW_CosSim_smpl) <- c(paste0("Robinson",RCsample),paste0("Women",LWsample))
```

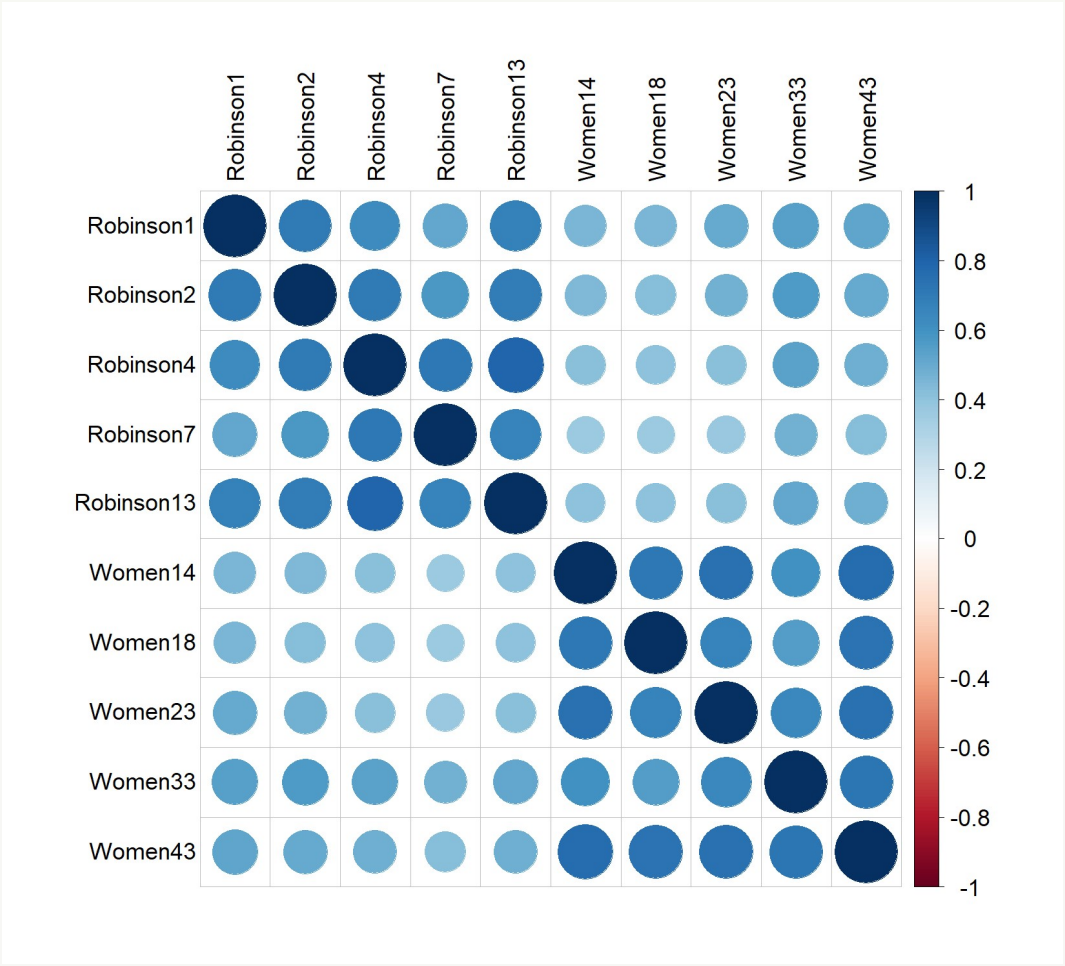
```
> RCLW_CosSim_smpl
           Robinson1 Robinson2 Robinson4 Robinson7 Robinson13   Women14
Robinson1  1.0000000 0.7056405 0.6277638 0.5171784  0.6751838 0.4590763
Robinson2  0.7056405 1.0000000 0.7008098 0.5755623  0.6959185 0.4526909
Robinson4  0.6277638 0.7008098 1.0000000 0.7150462  0.7928879 0.4143773
Robinson7  0.5171784 0.5755623 0.7150462 1.0000000  0.6644168 0.3664794
Robinson13 0.6751838 0.6959185 0.7928879 0.6644168  1.0000000 0.4046909
Women14    0.4590763 0.4526909 0.4143773 0.3664794  0.4046909 1.0000000
Women18    0.4637818 0.4364690 0.4113735 0.3742913  0.4140961 0.7132955
Women23    0.4982397 0.4710427 0.4120263 0.3719339  0.4079418 0.7544894
Women33    0.5358528 0.5537789 0.5307859 0.4689156  0.5088239 0.6305278
Women43    0.5280322 0.5041804 0.4891263 0.4287855  0.4871868 0.7551314
             Women18   Women23   Women33   Women43
Robinson1  0.4637818 0.4982397 0.5358528 0.5280322
Robinson2  0.4364690 0.4710427 0.5537789 0.5041804
Robinson4  0.4113735 0.4120263 0.5307859 0.4891263
Robinson7  0.3742913 0.3719339 0.4689156 0.4287855
Robinson13 0.4140961 0.4079418 0.5088239 0.4871868
Women14    0.7132955 0.7544894 0.6305278 0.7551314
Women18    1.0000000 0.6770986 0.5727019 0.7383179
Women23    0.6770986 1.0000000 0.6579654 0.7433836
Women33    0.5727019 0.6579654 1.0000000 0.7255801
Women43    0.7383179 0.7433836 0.7255801 1.0000000
```

2. 코사인 유사도를 이용한 문서의 분류

3 코사인 유사도의 시각화

- CosSim 행렬의 시각화를 통해 코사인 유사도를 보기 쉽게 표현
  - corrplot 패키지 내의 corrplot( ) 함수는 원래 상관행렬(correlation matrix)을 시각화하는 도구
    - → 상관행렬이 아니더라도 -1에서 1까지의 값을 가지는 행렬을 시각화하기 위해서도 사용 가능

```
> install.packages("corrplot")
> library(corrplot)
> corrplot(RCLW_CosSim_smpl)
```


Robinson1
Robinson2
Robinson4
Robinson7
Robinson13
Women14
Women18
Women23
Women33
Women43
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

실습하기

다음시간안내

13

텍스트 데이터의 통계적 분석(2)