自然言語処理入門

岸山 健 (31-187002)

Oct. 29, 2018

1 課題 1

まず与えられた文書 1–5 に含まれる単語 1–6 の単語頻度 tf(t,d) を求める． その単語頻度 tf(t,d) に加え，各単語に対して与えられた文書頻度 df(t) を表形式で変数 df に格納する．なお，文書の集合が含む文書数は N に格納する．

```
N <- 10000
df <- data.frame(
    word =c( '薬', '風邪', '熱', '喉', '胃', '消化'),
    df   =c(9030 ,   670 , 184 ,  27 , 428 ,   359 ),
    tf.d1=c(   2 ,     3 ,   0 ,   1 ,   0 ,     0 ),
    tf.d2=c(   1 ,     1 ,   1 ,   0 ,   0 ,     0 ),
    tf.d3=c(   2 ,     0 ,   1 ,   1 ,   0 ,     0 ),
    tf.d4=c(   4 ,     0 ,   0 ,   0 ,   3 ,     2 ),
    tf.d5=c(   1 ,     0 ,   0 ,   0 ,   1 ,     4 ))
df
##   word   df tf.d1 tf.d2 tf.d3 tf.d4 tf.d5
## 1   薬 9030     2     1     2     4     1
## 2 風邪  670     3     1     0     0     0
## 3   熱  184     0     1     1     0     0
## 4   喉   27     1     0     1     0     0
## 5   胃  428     0     0     0     3     1
## 6 消化  359     0     0     0     2     4
```

次に tf(t,d),d(t),N を引数に tf*idf を求める関数 tf.idf を定義する．そして tf.idf を文書ごとに適用し，各文書の各単語に対する tf*idf を求める．なお列は$で取得できるものとする．

```
tf.idf <- function(tf.d, df, N) tf.d*(log10(N/df)+1)

df$tf.idf.1 <- tf.idf(df$tf.d1, df$df, N)
df$tf.idf.2 <- tf.idf(df$tf.d2, df$df, N)
df$tf.idf.3 <- tf.idf(df$tf.d3, df$df, N)
df$tf.idf.4 <- tf.idf(df$tf.d4, df$df, N)
df$tf.idf.5 <- tf.idf(df$tf.d5, df$df, N)
```

```
##   word   df tf.d1 tf.d2 tf.d3 tf.d4 tf.d5
## 1   薬 9030     2     1     2     4     1
## 2 風邪  670     3     1     0     0     0
## 3   熱  184     0     1     1     0     0
## 4   喉   27     1     0     1     0     0
## 5   胃  428     0     0     0     3     1
## 6 消化  359     0     0     0     2     4
##   tf.idf.1 tf.idf.2 tf.idf.3 tf.idf.4 tf.idf.5
## 1 2.088624 1.044312 2.088624 4.177249 1.044312
## 2 6.521776 2.173925 0.000000 0.000000 0.000000
## 3 0.000000 2.735182 2.735182 0.000000 0.000000
## 4 3.568636 0.000000 3.568636 0.000000 0.000000
## 5 0.000000 0.000000 0.000000 7.105669 2.368556
## 6 0.000000 0.000000 0.000000 4.889811 9.779622
```

上の計算にしたがい, 各単語に対する tf*idf 値は以下のとおりとなる．列名は文書名を示すので tf.idf.1 は文書位置に対する各単語の tf*idf 値となっている． つまり tf.idf.1 と薬が交差する点は，文書 1 の「薬」という単語に対する tf*idf 値である．

```
##       tf.idf.1 tf.idf.2 tf.idf.3 tf.idf.4 tf.idf.5
##   薬 2.088624 1.044312 2.088624 4.177249 1.044312
## 風邪 6.521776 2.173925 0.000000 0.000000 0.000000
##   熱 0.000000 2.735182 2.735182 0.000000 0.000000
##   喉 3.568636 0.000000 3.568636 0.000000 0.000000
##   胃 0.000000 0.000000 0.000000 7.105669 2.368556
## 消化 0.000000 0.000000 0.000000 4.889811 9.779622
```

2 課題 2

次に上で求めた文書の各単語に対する tf*idf をその文書を表象する重みとみなし文書特徴表現とする．したがって，以下の列をそのまま各文書の特徴とする．

```
##       tf.idf.1 tf.idf.2 tf.idf.3 tf.idf.4 tf.idf.5
##   薬 2.088624 1.044312 2.088624 4.177249 1.044312
## 風邪 6.521776 2.173925 0.000000 0.000000 0.000000
##   熱 0.000000 2.735182 2.735182 0.000000 0.000000
##   喉 3.568636 0.000000 3.568636 0.000000 0.000000
##   胃 0.000000 0.000000 0.000000 7.105669 2.368556
## 消化 0.000000 0.000000 0.000000 4.889811 9.779622
```

これに対し，検索質問の<薬:1.0, 風邪:3.0, 熱:2.0>と <薬:1.0, 胃:2.0>との類似度をそれぞれ求め，並び替える．ベクトル同士の類似度には余弦尺度が使えるので 必要な関数を定義する．まず %ip% は 1. 引数のベクトル同士を 2 列に格納し， 2. それらの行ごとで積を取り， 3. その結果の和をとって返す中置関数である．また，D はベクトルの長さを返す関数である．余弦尺度を求める関数 Sim.c は検索質問をまず引数に取り，その

ベクトルに対する類似度を返す関数を返す.

```
# l と r の内積を返す中置関数を定義
'%ip%' <- function (l,r) {
    # 入力のベクトル l,r を列に格納
    tmp <- data.frame(l=l,r=r)
    # l.r という列名の各列に l*r を格納
    tmp$l.r <- tmp$l * tmp$r
    # l.r の和を取る
    sum(tmp$l.r)}


# ベクトル (Ws) の長さを返す関数を定義
D <- function(Ws) sqrt(sum(Ws ** 2))

# 第一引数に q(uestion), 第二引数に d(ocument vector)
Sim.c <- function(q) function(d) q %ip% d / (D(d) * D(q))
```

以上の関数の動作を確認する．講義内で扱われた検索質問<2,1,3,0>を q0，対象となった文書のベクトル<4,3,0,5>を q0 とする．それらを関数に入れることで余弦尺度が求まる．なお，単語の情報は各列の index からアクセスできる.

```
# 文書 1 = <2,1,3,0> と 質問=<4,3,0,5> で動作確認
d0 <- c(2,1,3,0)
q0 <- c(4,3,0,5)

Sim.c(q0)(d0)
## [1] 0.4157609
```

最初の検索質問は<薬:1.0, 風邪:3.0, 熱:2.0>であった． まずはこの質問を q1 に格納し，Sim.c.q1 にこの質問との類似度を返す関数を格納する．そして df.rep という表に検索質問と同じ次元を持ったベクトルを格納する．つまり，薬は 1 行目，風邪は 2 行目，熱は 3 行目にあるため，c(1,2,3) と index することでアクセスでき， それを列に格納する.

```
# 検索質問: 薬, 風邪, 熱
q1 = c(1,3,2)
# q1 との類似度を返す関数
Sim.c.q1 <- Sim.c(q1)

# 1 薬, 2 風邪, 3 熱, 4 のど, 5 胃, 6 消化のベクトル
df $ tf.idf.1
## [1] 2.088624 6.521776 0.000000 3.568636 0.000000 0.000000

# 1 薬, 2 風邪, 3 熱のベクトルには c(1,2,3) でアクセスできる.
# index は 1 スタート
df.rep <- data.frame(
```

```
    d1 = df $ tf.idf.1 [c(1,2,3)],
    d2 = df $ tf.idf.2 [c(1,2,3)],
    d3 = df $ tf.idf.3 [c(1,2,3)],
    d4 = df $ tf.idf.4 [c(1,2,3)],
    d5 = df $ tf.idf.5 [c(1,2,3)])
df.rep
#         d1       d2       d3       d4       d5
# 1 2.088624 1.044312 2.088624 4.177249 1.044312
# 2 6.521776 2.173925 0.000000 0.000000 0.000000
# 3 0.000000 2.735182 2.735182 0.000000 0.000000
```

この状態において df.rep$d1 は 文書 1 のベクトルが<薬:2.088624, 風邪:6.521776, 熱:0.0> であることが d1 の行の 1,2,3 行目を見るとわかる．この列をベクトルとして取得し，先ほどのの Sim.c.q1 に与えれば q1 との類似度が求まる．それを df.sim という表に格納する

```
df.sim <- data.frame(
    d1 = Sim.c.q1(df.rep $ d1) ,
    d2 = Sim.c.q1(df.rep $ d2) ,
    d3 = Sim.c.q1(df.rep $ d3) ,
    d4 = Sim.c.q1(df.rep $ d4) ,
    d5 = Sim.c.q1(df.rep $ d5) )

df.sim
##          d1       d2        d3        d4        d5
## 1 0.8450952 0.955446 0.5870273 0.2672612 0.2672612

df.sim[order(df.sim[1,],decreasing=T)]
##         d2        d1        d3        d4        d5
## 1 0.955446 0.8450952 0.5870273 0.2672612 0.2672612
```

表に格納した後は降順にして出力する． その結果，文書 2, 文書 1, 文書 3, そして文書 4,5 の順で 似ていたことが分かる．

次の検索質問は<薬:1.0, 胃:2.0>であったので，同様の手順を繰り返す．

```
# 検索質問：薬，胃
q2 = c(1,2)
# q1 との類似度を返す関数
Sim.c.q2 <- Sim.c(q2)

# 薬，胃のベクトルには c(1,5) でアクセスできる．
# index は 1 スタート
df.rep <- data.frame(
    d1 = df $ tf.idf.1 [c(1,5)],
    d2 = df $ tf.idf.2 [c(1,5)],
    d3 = df $ tf.idf.3 [c(1,5)],
```

```
    d4 = df $ tf.idf.4 [c(1,5)],
    d5 = df $ tf.idf.5 [c(1,5)])
df.rep
##        d1       d2       d3       d4       d5
##1 2.088624 1.044312 2.088624 4.177249 1.044312
##2 0.000000 0.000000 0.000000 7.105669 2.368556
```

この状態において df.rep$d1 は 文書 1 のベクトルが<薬:2.088624, 胃:0.0> であることが d1 の行の 1,2,3 行目を見るとわかる．この列をベクトルとして取得し，先ほどのの Sim.c.q2 に与えれば q2 との類似度が求まる．それを df.sim という表に格納する

```
df.sim <- data.frame(
    d1 = Sim.c.q2(df.rep $ d1) ,
    d2 = Sim.c.q2(df.rep $ d2) ,
    d3 = Sim.c.q2(df.rep $ d3) ,
    d4 = Sim.c.q2(df.rep $ d4) ,
    d5 = Sim.c.q2(df.rep $ d5) )

df.sim
##         d1        d2        d3        d4        d5
##1 0.4472136 0.4472136 0.4472136 0.9977018 0.9988298

df.sim[order(df.sim[1,],decreasing=T)]
##         d5        d4        d1        d2        d3
##1 0.9988298 0.9977018 0.4472136 0.4472136 0.4472136
```

したがって，文書 5, 文書 4, 文書 1, 文書 2,3 の順で類似度が検索質問 2 に近いことが分かった．