自然言語処理入門

岸山 健 (31-187002)

Nov. 5, 2018

1 課題 1

まず与えられた文書 1–6 に含まれる単語の単語頻度 tf をデータフレーム df に与える[*1]．そして各文書ごとの類似度を tf 値を用いて求める．

```
df <- data.frame(
#    word =c( 'ウィルス', 'エイズ ', '肝炎', '感染', 'コンピュータ',
#             '被害','ファイル','米国','メール'),
    tf.d1=c(4,0,0,3,1,1,0,1,2),
    tf.d2=c(3,0,0,3,1,1,1,0,2),
    tf.d3=c(2,0,0,1,2,1,2,0,0),
    tf.d4=c(2,1,1,2,0,1,0,3,0),
    tf.d5=c(1,0,2,2,0,1,0,0,0),
    tf.d6=c(1,3,0,3,0,1,0,0,0))
```

類似度を求めるために 各文書から 2 つ選択する組み合わせを docs.C.2 リストとして作成する．先ほどの df のカラム名は文書名を示すので，df のカラム名を colnames 関数で取得し，それらを 2 つ選択する組み合わせを combn 関数で取得する．そして t 関数で転値行列をつくりデータフレームとして docs.C.2 に与える．

この docs.C.2 は行番号 i で各「文書の組み合わせ」にアクセスできるため i を 1 から行数を回して「文書の組み合わせ」の類似度をコサイン類似度で得ることができる．行ごとに得た「文書の組み合わせ」の類似度は docs.C.2 の sim という列に割り当てる．

```
library(dplyr)
library(purrr)

# 組み合わせの行列を作る
combn(colnames(df), 2)
docs.C.2 <- as.data.frame(t(combn(colnames(df), 2)))
docs.C.2$sim = 1:nrow(docs.C.2) %>% map_dbl(function(i)
             Sim.c(unlist(select(df,as.character(docs.C.2[i,]$V1))) ,
                   unlist(select(df,as.character(docs.C.2[i,]$V2)))))
docs.C.2
```

[*1] 回答の一部には GNU R を用いた．

```
##       V1    V2       sim
## 1  tf.d1 tf.d2 0.9545942
## 2  tf.d1 tf.d3 0.6614378
## 3  tf.d1 tf.d4 0.7115125
## 4  tf.d1 tf.d5 0.6149187
## 5  tf.d1 tf.d6 0.5533986
## 6  tf.d2 tf.d3 0.7483315
## 7  tf.d2 tf.d4 0.5813777
## 8  tf.d2 tf.d5 0.6324555
## 9  tf.d2 tf.d6 0.5813777
## 10 tf.d3 tf.d4 0.4183300
## 11 tf.d3 tf.d5 0.4225771
## 12 tf.d3 tf.d6 0.3585686
## 13 tf.d4 tf.d5 0.6363961
## 14 tf.d4 tf.d6 0.6000000
## 15 tf.d5 tf.d6 0.5656854
```

上のデータフレームを表にすると以下のとおりとなり，最も似ているのは文書 1 と文書 2 である (コサイン類似度=0.95)．これらを統合した際にクラスターが発生するが，そのクラスターと他の文書 (あるいは他のクラスター) の類似度は複数の方法で求められる．本課題では完全リンク法と単一リンク法を用いてクラスタリングする．そしてそれぞれで作成したクラスターに対し 樹形図を作成する．

	tf.d2	tf.d3	tf.d4	tf.d5	tf.d6
tf.d1	0.954	0.661	0.711	0.614	0.553
tf.d2		0.748	0.581	0.632	0.581
tf.d3			0.418	0.422	0.358
tf.d4				0.636	0.600
tf.d5					0.565

1.1 完全リンク法

まず完全リンク法でクラスタリングする．はじめに文書 1 と文書 2 を結合しクラスターを作成し，そのクラスターと他文書の類似度を測る．完全リンク法は類似度の最小値をとるため， 以下のような表を作成できる．

	tf.d3	tf.d4	tf.d5	tf.d6
tf.d1+2	0.661	0.581	0.614	0.553
tf.d3		0.418	0.422	0.358
tf.d4			0.636	0.600
tf.d5				0.565

文書 1,2 が形成するクラスターと他の文書との類似度は最小となるものを選択している．文書 3 と最も離れているのは文書 1(0.661) なので，これが文書 1,2 のクラスターと文書 3 のクラスターの類似度となる．そして文

書 4 では文書 2 の 0.581, 文書 5 では文書 1 との 0.614, 文書 6 では文書 1 の 0.553 が新しい類似度となる．

次は 1,2 のクラスターと 3 を結合する． そして先ほど同様，文書 1,2,3 と他のクラスターの類似度を見て，もっとも離れている値を新しい類似度とする．

	tf.d4	tf.d5	tf.d6
tf.d1+2+3	0.418	0.422	0.358
tf.d4		0.636	0.600
tf.d5			0.565

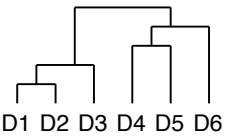
次にクラスターを形成するのは文書 4 と文書 5 である．文書 4,5 と文書 1,2,3 の組み合わせをそれぞれ見て，最も離れている値を選択する．すると 0.418 が文書 4,5–文書 1,2,3 間の類似度となることが分かる．また文書 4,5 と文書 6 は 0.60 か 0.56 の選択肢があり， 後者が選択される．

	tf.d4+5	tf.d6
tf.d1+2+3	0.418	0.358
tf.d4+5		0.565

次は文書 4,5 と文書 6 をくっつけ，これらの類似度は 0.358 となる．

	tf.d4+5+6
tf.d1+2+3	0.358

以上の手順で作成した階層的クラスターからは 以下の樹形樹が書ける．高さは結合した順番を示している．



1.2 単一リンク法

	tf.d2	tf.d3	tf.d4	tf.d5	tf.d6
tf.d1	0.954	0.661	0.711	0.614	0.553
tf.d2		0.748	0.581	0.632	0.581
tf.d3			0.418	0.422	0.358
tf.d4				0.636	0.600
tf.d5					0.565

次は単一リンク法でクラスタリングを行なう．この手法ではクラスタ間の類似度は 最も似ているクラスタ間の値となる．したがって，文書 1,2 が形成するクラスターと 他の文書の類似度を比較する際はもっとも高い値

をそのクラスターと文書間の新しい類似度とする．

	tf.d3	tf.d4	tf.d5	tf.d6
tf.d1+2	0.748	0.711	0.632	0.581
tf.d3		0.418	0.422	0.358
tf.d4			0.636	0.600
tf.d5				0.565

したがって文書 1,2 のクラスターと文書 3 では 0.74, 文書 4 では 0.71, 文書 5 では 0.63, 文書 6 では 0.58 が選択され新しい類似度となる． 次は文書 1,2 と文書 3 の類似度が最も高いため，これらを結合し新しいクラスターとする．

	tf.d4	tf.d5	tf.d6
tf.d1+2+3	0.711	0.632	0.581
tf.d4		0.636	0.600
tf.d5			0.565

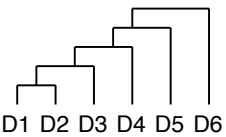
次に結合するのは文書 123 と文書 4 となる．先ほどの例では文書 4 と統合されるのは文書 5 であったが，単一リンク法が作る新しいクラスターの類似度の方法により，高い値が維持されて結合し続けるようにみえる．

	tf.d5	tf.d6
tf.d1+2+3+4	0.636	0.600
tf.d5		0.565

そして最後は以下となる．

	tf.d6
tf.d1+2+3+4+5	0.600

以上の手順で作成した階層的クラスターからは 以下の樹形樹が書ける．高さは結合した順番を示している．完全リンク法と比較すると，伸びるような振る舞いでクラスターを形成していることが分かる．



1.3 前回つくったコサイン類似度を求める関数

```
# 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))

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