IS 622 Week 11 Homework

Ben Arancibia

November 5, 2015

9.3.1

Figure 9.8 is a utility matrix, representing the ratings, on a 1–5 star scale, of eight items, a through h, by three users A, B, and C. Compute the following from the data of this matrix.

	a	b	c	d	e	f	g	h
A	4	5		5	1		3	2
B		3	4	3	1	2	1	
C	2		1	3		4	5	3

(a) Treating the utility matrix as boolean, compute the Jaccard distance between each pair of users.

```
df <- data.frame(a = c(4, NA, 2),
                 b = c(5, 3, NA),
                 c = c(NA, 4, 1),
                 d = c(5, 3, 3),
                 e = c(1, 1, NA),
                 f = c(NA, 2, 4),
                 g = c(3, 1, 5),
                 h = c(2, NA, 3))

rownames(df) <- c("A", "B", "C")
cn <- colnames(df)
user.pairs <- as.data.frame(t(combn(rownames(df), 2)))
colnames(user.pairs) <- c("User1", "User2")

jaccard <- function(v1, v2) {
  length(intersect(v1, v2)) / length(union(v1, v2))
}
inset <- function(j) {!is.na(j) & as.logical(j)}
jd <- user.pairs
jd["Jaccard.Distance"] <-
  apply(user.pairs, 1, function(i) {
    s1 <- cn[sapply(df[i[1], ], inset)]
    s2 <- cn[sapply(df[i[2], ], inset)]
    jaccard(s1, s2)
})

jd
```

```
##   User1 User2 Jaccard.Distance
```

```
## 1     A     B              0.5
## 2     A     C              0.5
## 3     B     C              0.5
```

(b) Repeat Part (a), but use the cosine distance.

```
len <- function(v) { sqrt(sum(v**2)) }
cosine.dist <- function(v1, v2) {
  v1[is.na(v1)] <- 0
  v2[is.na(v2)] <- 0
  (as.numeric(v1) %*% as.numeric(v2)) / (len(v1) * len(v2))
}

# Apply distance function
cd <- user.pairs
cd["Cosine.Distance"] <-
  apply(user.pairs, 1, function(i) {
    cosine.dist(df[i[1], ], df[i[2], ])
})

# Format output
cd
```

```
##   User1 User2 Cosine.Distance
## 1     A     B       0.6010408
## 2     A     C       0.6149187
## 3     B     C       0.5138701
```

(c) Treat ratings of 3, 4, and 5 as 1 and 1, 2, and blank as 0. Compute the Jaccard distance between each pair of users.

```
binrating <- function(i) {ifelse(i %in% c(3, 4, 5), TRUE, FALSE)}

# Apply distance function
jd.c <- user.pairs
jd.c["Jaccard.Distance"] <-
  apply(user.pairs, 1, function(i) {
    s1 <- cn[sapply(df[i[1], ], binrating)]
    s2 <- cn[sapply(df[i[2], ], binrating)]
    jaccard(s1, s2)
})

jd.c
```

```
##   User1 User2 Jaccard.Distance
## 1     A     B        0.4000000
## 2     A     C        0.3333333
## 3     B     C        0.1666667
```

(d) Repeat Part (c), but use the cosine distance.

```
cd.d <- user.pairs
cd.d["Cosine.Distance"] <-
  apply(user.pairs, 1, function(i) {
    v1 <- binrating(df[i[1], ])
    v2 <- binrating(df[i[2], ])
    cosine.dist(v1, v2)
  })

# Format output
cd.d
```

```
##   User1 User2 Cosine.Distance
## 1     A     B       0.5773503
## 2     A     C       0.5000000
## 3     B     C       0.2886751
```

(e) Normalize th ematrix by subtracting from each nonblank entry the average value for its user.

```
df.normalized <- t(apply(df, 1, function(i) {
  i - mean(i, na.rm=TRUE)
}))

df.normalized
```

```
##            a         b         c         d         e          f          g
## A  0.6666667 1.6666667        NA 1.6666667 -2.333333         NA -0.3333333
## B         NA 0.6666667  1.666667 0.6666667 -1.333333 -0.3333333 -1.3333333
## C -1.0000000        NA -2.000000 0.0000000        NA  1.0000000  2.0000000
##           h
## A -1.333333
## B        NA
## C  0.000000
```

(f) Using the normalized matrix from Part (e), compute the cosine distance between each pair of users.

```
df.normalized[is.na(df.normalized)] <- 0

# Apply distance function
cd.f <- user.pairs
cd.f["Cosine.Distance"] <-
  apply(user.pairs, 1, function(i) {
    v1 <- df.normalized[i[1], ]
    v2 <- df.normalized[i[2], ]
    cosine.dist(v1, v2)
  })

# Format output
cd.f
```

```
##   User1 User2 Cosine.Distance
## 1     A     B       0.5843065
## 2     A     C      -0.1154701
## 3     B     C      -0.7395740
```