Wk 11 Exercises

Aaron Palumbo

November 3, 2015

Contents

9.3.1 (section 9.3.4)

Table 1: Figure 9.8: A utility matrix for exercises

	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

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.

```
# ############## #
# Utility Matrix #
# ############## #
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 Combinations #
# ################# #
user.pairs <- as.data.frame(t(combn(rownames(df), 2)))
colnames(user.pairs) <- c("User1", "User2")
```

9.3.1 (a)

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

```
# ################ #
# Jaccard Distance #
# ################ #

# Distance function
jaccard.dist <- function(v1, v2) {
  length(intersect(v1, v2)) / length(union(v1, v2))
}

# Return boolean to determine set membership
inset <- function(j) {!is.na(j) & as.logical(j)}

# Apply distance function
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.dist(s1, s2)
})

# Format output
knitr::kable(jd, digits=2)
```

User1	User2	Jaccard.Distance
A	B	0.5
A	C	0.5
B	C	0.5

9.3.1 (b)

Repeat part (a), but use the cosine distance.

```
# ############### #
# Cosine Distance #
# ############### #

# Distance function
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
knitr::kable(cd, digits=2)
```

User1	User2	Cosine.Distance
A	B	0.60
A	C	0.61
B	C	0.51

9.3.1 (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.

```
# Transform Data
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.dist(s1, s2)
})

# Format output
knitr::kable(jd.c, digits=2)
```

User1	User2	Jaccard.Distance
A	B	0.40
A	C	0.33
B	C	0.17

9.3.1 (d)

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

```
# Apply distance function
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
knitr::kable(cd.d, digits=2)
```

User1	User2	Cosine.Distance
A	B	0.58
A	C	0.50
B	C	0.29

9.3.1 (e)

Normalize the matrix by subtracting from each non blank entry the average value for its user.

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

knitr::kable(df.normalized, digits=2)
```

	a	b	c	d	e	f	g	h
A	0.67	1.67	NA	1.67	-2.33	NA	-0.33	-1.33
B	NA	0.67	1.67	0.67	-1.33	-0.33	-1.33	NA
C	-1.00	NA	-2.00	0.00	NA	1.00	2.00	0.00

9.3.1 (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
knitr::kable(cd.f, digits=2)
```

User1	User2	Cosine.Distance
A	B	0.58
A	C	-0.12
B	C	-0.74