Data Mining HW 3

Ethen Liu, Sai Haran, Sophia Hoffman, Annie Didier, Arindam Bhattacharya

January 23, 2017

Question 1

a)

```
X = matrix(c(1,3,5,0,1, 2,4,4,2,3, 3,5,3,4,5), nrow=5)
x_tran_x = t(X) %*% X
x_x_tran = X %*% t(X)
x_tran_x
```

```
##      [,1] [,2] [,3]
## [1,]   36   37   38
## [2,]   37   49   61
## [3,]   38   61   84
```

```
x_x_tran
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   14   26   22   16   22
## [2,]   26   50   46   28   40
## [3,]   22   46   50   20   32
## [4,]   16   28   20   20   26
## [5,]   22   40   32   26   35
```

b)

```
ei_x_tran_x = eigen(x_tran_x)
ei_x_x_tran = eigen(x_x_tran)
ei_x_tran_x
```

```
## $values
## [1] 1.535670e+02 1.543300e+01 1.421085e-14
##
## $vectors
##            [,1]       [,2]       [,3]
## [1,] -0.4092828  0.8159785  0.4082483
## [2,] -0.5634593  0.1258846 -0.8164966
## [3,] -0.7176358 -0.5642094  0.4082483
```

```
ei_x_x_tran
```

```
## $values
## [1]  1.535670e+02  1.543300e+01  1.998401e-14  2.848207e-15 -4.440892e-16
##
## $vectors
##            [,1]       [,2]       [,3]        [,4]        [,5]
## [1,] -0.2976957  0.1590639  0.0000000  0.94131607  0.00000000
## [2,] -0.5705086 -0.0332003  0.7978581 -0.17481584 -0.07924371
## [3,] -0.5207430 -0.7358566 -0.4209721 -0.04034212 -0.09217818
## [4,] -0.3225785  0.5103921 -0.3207010 -0.18826321 -0.70508927
## [5,] -0.4589849  0.4142600 -0.2887141 -0.21515796  0.69862203
```

c)

```r
A = ei_x_tran_x$vectors%*%diag(ei_x_tran_x$values)%*% t(ei_x_tran_x$vectors)
A
```

```
##      [,1] [,2] [,3]
## [1,]   36   37   38
## [2,]   37   49   61
## [3,]   38   61   84
```

The output is the same as 2a.

d)

```r
svd_x = svd(X)
#Compare to b
svd_x$v
```

```
##            [,1]       [,2]       [,3]
## [1,] -0.4092828 -0.8159785 -0.4082483
## [2,] -0.5634593 -0.1258846  0.8164966
## [3,] -0.7176358  0.5642094 -0.4082483
```

```r
ei_x_tran_x$vectors
```

```
##            [,1]       [,2]       [,3]
## [1,] -0.4092828  0.8159785  0.4082483
## [2,] -0.5634593  0.1258846 -0.8164966
## [3,] -0.7176358 -0.5642094  0.4082483
```

```r
svd_x$u
```

```
##            [,1]       [,2]        [,3]
## [1,] -0.2976957  0.1590639  0.90607622
## [2,] -0.5705086 -0.0332003  0.03827317
## [3,] -0.5207430 -0.7358566 -0.13315536
## [4,] -0.3225785  0.5103921 -0.18363343
## [5,] -0.4589849  0.4142600 -0.35511895
```

```r
ei_x_x_tran$vectors
```

```
##            [,1]       [,2]       [,3]        [,4]        [,5]
## [1,] -0.2976957  0.1590639  0.0000000  0.94131607  0.00000000
## [2,] -0.5705086 -0.0332003  0.7978581 -0.17481584 -0.07924371
## [3,] -0.5207430 -0.7358566 -0.4209721 -0.04034212 -0.09217818
## [4,] -0.3225785  0.5103921 -0.3207010 -0.18826321 -0.70508927
## [5,] -0.4589849  0.4142600 -0.2887141 -0.21515796  0.69862203
```

```r
(svd_x$d)^2
```

```
## [1] 1.53567e+02 1.54330e+01 3.28692e-31
```

```r
(ei_x_tran_x$values)
```

```
## [1] 1.535670e+02 1.543300e+01 1.421085e-14
```

```r
ei_x_x_tran$values
```

```
## [1]  1.535670e+02  1.543300e+01  1.998401e-14  2.848207e-15 -4.440892e-16
```

We can see that V gives the eigenvectors of X'X and that the first two columns of U correspond to the first two eigenvectors of XX' since the rank of X is 2.

e)

```r
ULV = svd_x$u %*% diag(svd_x$d) %*% t(svd_x$v)
ULV
```

```
##              [,1] [,2] [,3]
## [1,] 1.000000e+00    2    3
## [2,] 3.000000e+00    4    5
## [3,] 5.000000e+00    4    3
## [4,] 4.870696e-16    2    4
## [5,] 1.000000e+00    3    5
```

```r
X
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    3    4    5
## [3,]    5    4    3
## [4,]    0    2    4
## [5,]    1    3    5
```

f)

```r
#set the third value to 0 since X has rank 2
svd_x_rank2 = svd_x$d[1:2]
#set the smallest eigenvalue to 0 to do the 1-d projection
svd_x_oned = svd_x$d
svd_x_oned[2:3] = 0
svd_u_one = svd_x$u
svd_u_one[,2:3] = 0
svd_v_one = svd_x$v
svd_v_one[,2:3] = 0
xhat = svd_u_one %*% diag(svd_x_oned) %*% t(svd_v_one)
xhat
```

```
##          [,1]     [,2]     [,3]
## [1,] 1.509889 2.078663 2.647437
## [2,] 2.893574 3.983581 5.073588
## [3,] 2.641167 3.636093 4.631018
## [4,] 1.636093 2.252407 2.868722
## [5,] 2.327935 3.204866 4.081797
```

xhat is very close to x, so the 1D estimate is a good approximation.

g)

```r
ssxhat = sum(xhat^2)
ssxhat
```

```
## [1] 153.567
```

```r
sum(svd_x$d^2)
```

```
## [1] 169
```

The Frobenius norm of xhat is the same as the sum of squares of the singular values.

h)

```r
SSE = sum((X-xhat)^2)
SSE
```

```
## [1] 15.433
```

```r
svd_x$d[2] ^ 2
```

```
## [1] 15.433
```

It is equal to the square of the k + 1 singular values, where k is the number of the top k singular value we chose to compute the Xhat approximation.

i)

```r
energy = svd_x_oned^2/(sum(svd_x$d^2))
energy
```

```
## [1] 0.9086805 0.0000000 0.0000000
```

Question 2

```r
library(data.table)
setwd("/Users/ethen/Desktop/northwestern/winter/MSIA 421 Data Mining/hw3")
theater <- fread('theater.csv')
theater <- theater[ , -c(3, 5, 7, 10), with = FALSE ]

# the theater subset data will be used in question e
theater_subset <- theater[ , .(dinner, play) ]
theater[ , c('dinner', 'play', 'age', 'educ', 'income', 'cnty') := NULL ]
```

a)

```r
alpha_theater1 <- psych::alpha(theater, check.keys = TRUE)
alpha_theater1
```

```
##
## Reliability analysis
## Call: psych::alpha(x = theater, check.keys = TRUE)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean  sd
##       0.93      0.94    0.94      0.59  15 0.0019  5.2 1.2
##
##  lower alpha upper     95% confidence boundaries
## 0.93 0.93 0.94
##
##  Reliability if an item is dropped:
##                  raw_alpha std.alpha G6(smc) average_r S/N alpha se
## stimulate             0.92      0.93    0.93      0.58  12   0.0022
## dislike-              0.94      0.94    0.94      0.64  16   0.0017
## fun                   0.92      0.92    0.92      0.57  12   0.0023
## irritate-             0.93      0.93    0.93      0.59  13   0.0021
## bad-                  0.92      0.93    0.93      0.58  12   0.0022
## timewellspent         0.93      0.93    0.93      0.59  13   0.0021
## exciting              0.92      0.92    0.93      0.58  12   0.0022
## noteduc-              0.94      0.94    0.94      0.64  16   0.0018
## comfortable           0.92      0.93    0.93      0.58  12   0.0022
## cannotappreciate-     0.93      0.93    0.93      0.59  13   0.0021
##
```

```
##  Item statistics
##                      n raw.r std.r r.cor r.drop mean  sd
## stimulate         2692  0.86  0.86  0.85   0.82  5.3 1.7
## dislike-          2692  0.61  0.60  0.53   0.51  4.8 1.7
## fun               2692  0.88  0.88  0.89   0.85  5.3 1.6
## irritate-         2692  0.81  0.82  0.79   0.76  5.2 1.5
## bad-              2692  0.86  0.86  0.85   0.82  5.5 1.4
## timewellspent     2692  0.81  0.80  0.78   0.75  5.1 1.6
## exciting          2692  0.87  0.87  0.86   0.83  5.2 1.6
## noteduc-          2692  0.59  0.60  0.52   0.50  4.9 1.5
## comfortable       2692  0.85  0.85  0.84   0.81  5.2 1.5
## cannotappreciate- 2692  0.83  0.83  0.81   0.78  5.6 1.5
##
## Non missing response frequency for each item
##                     1    2    3    4    5    6    7 miss
## stimulate        0.05 0.03 0.03 0.18 0.16 0.26 0.28    0
## dislike          0.19 0.22 0.12 0.28 0.07 0.06 0.05    0
## fun              0.04 0.03 0.04 0.18 0.16 0.27 0.28    0
## irritate         0.23 0.27 0.16 0.24 0.04 0.03 0.03    0
## bad              0.30 0.27 0.15 0.23 0.02 0.02 0.02    0
## timewellspent    0.03 0.04 0.06 0.26 0.12 0.23 0.25    0
## exciting         0.04 0.03 0.04 0.23 0.18 0.25 0.24    0
## noteduc          0.20 0.21 0.14 0.32 0.06 0.04 0.03    0
## comfortable      0.03 0.03 0.04 0.26 0.16 0.26 0.23    0
## cannotappreciate 0.37 0.26 0.12 0.17 0.03 0.03 0.03    0
```

b)

```
pca1 <- prcomp(theater)
pca1$sdev
```

```
##  [1] 3.9578487 1.4705079 1.3525016 1.0562171 0.9577743 0.8441564 0.8021089
##  [8] 0.7436424 0.6966951 0.6110656
```

4 eigenvalues are greater than 1.

```
pca1$rotation[, 1]
```

```
##        stimulate          dislike              fun         irritate
##       -0.3646057        0.2497032       -0.3605609        0.3031613
##              bad    timewellspent         exciting          noteduc
##        0.3078863       -0.3340471       -0.3462710        0.2200925
##      comfortable cannotappreciate
##       -0.3250952        0.3199804
```

noteduc, dislike have smaller first loading vectors when it comes to magnitude.

c)

```
# after removing noteduc and dislike,
# the drop of reliability remains the same for
# every single variable that we dropped
theater[ , c('dislike', 'noteduc') := NULL ]
alpha_theater2 <- psych::alpha(theater, check.keys = TRUE)
```

```
## Warning in psych::alpha(theater, check.keys = TRUE): Some items were negatively correlated with total
##  This is indicated by a negative sign for the variable name.
```

```r
alpha_theater2
```

```r
# and the explained ratio of the first loading vector is larger
pca2 <- prcomp(theater)
summary(pca1)
```

```
## Importance of components:
##                           PC1     PC2    PC3     PC4     PC5     PC6
## Standard deviation     3.9578 1.47051 1.3525 1.05622 0.95777 0.84416
## Proportion of Variance 0.6405 0.08842 0.0748 0.04561 0.03751 0.02914
## Cumulative Proportion  0.6405 0.72891 0.8037 0.84932 0.88683 0.91597
##                            PC7     PC8     PC9    PC10
## Standard deviation     0.80211 0.74364 0.69670 0.61107
## Proportion of Variance 0.02631 0.02261 0.01985 0.01527
## Cumulative Proportion  0.94227 0.96489 0.98473 1.00000
```

```r
summary(pca2)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     3.7609 1.16042 0.96781 0.85433 0.80384 0.74448
## Proportion of Variance 0.7358 0.07005 0.04873 0.03797 0.03362 0.02883
## Cumulative Proportion  0.7358 0.80589 0.85462 0.89259 0.92621 0.95504
##                            PC7    PC8
## Standard deviation     0.69818 0.6138
## Proportion of Variance 0.02536 0.0196
## Cumulative Proportion  0.98040 1.0000
```

d)

```r
# reverse code the reversed coded columns
reverse_col <- c('irritate', 'bad', 'cannotappreciate')
max_score <- 7
theater[ , (reverse_col) := lapply(.SD, function(col) {
    (max_score + 1) - col
}), .SDcols = reverse_col ]

# compute attitude variable
theater_subset[ , attitude := rowSums(theater) / ncol(theater) ]
```

```r
theater_subset
```

```
##       dinner play attitude
##    1:      2    4    6.625
##    2:      0    0    5.125
##    3:      2    0    5.875
##    4:      2    0    5.750
##    5:      2    0    6.875
##   ---
## 2688:      0    0    4.750
## 2689:      0    0    6.500
## 2690:      0    0    4.000
## 2691:      0    1    5.625
## 2692:      0    1    6.875
```

e)

```r
# there are NA values in theater subset, we tried dropping the
# NA values and setting the NA values to 0 and obtained similar result
theater_subset1 <- theater_subset[ complete.cases(theater_subset), ]
theater_subset1[ , new_feature := log(dinner + play + 1) ]

theater_subset[ is.na(dinner), dinner := 0 ]
theater_subset[ is.na(play), play := 0 ]
theater_subset[ , new_feature := log(dinner + play + 1) ]
```

```r
with( theater_subset1, cor(new_feature, attitude, method = 'pearson') )
```

```
## [1] 0.2729852
```

```r
with( theater_subset, cor(new_feature, attitude, method = 'pearson') )
```

```
## [1] 0.274381
```

f)

The sum of dinner and play is not within the normal range of 1 to 7, by taking the log, we transform it to a narrower and more comparable range.

Question 3

```r
music <- read.csv("music.csv")
retain_cols <- c('V28', 'V29', 'V30', 'V31', 'V32', 'V33',
                 'V34', 'V43', 'V46', 'V47', 'V50', 'V52')
music <- music[, retain_cols]
```

a)

```r
alpha_music <- psych::alpha(music)
alpha_music
```

```
##
## Reliability analysis
## Call: psych::alpha(x = music)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean sd
##       0.94      0.94    0.94      0.58  16 0.0024  2.8  1
##
##  lower alpha upper     95% confidence boundaries
## 0.94 0.94 0.95
##
##  Reliability if an item is dropped:
##     raw_alpha std.alpha G6(smc) average_r S/N alpha se
## V28      0.94      0.94    0.94      0.57  14   0.0027
## V29      0.94      0.94    0.94      0.57  15   0.0026
## V30      0.94      0.94    0.94      0.57  15   0.0026
## V31      0.94      0.94    0.94      0.60  16   0.0024
## V32      0.93      0.93    0.94      0.56  14   0.0027
## V33      0.94      0.94    0.94      0.57  15   0.0026
## V34      0.94      0.94    0.94      0.58  15   0.0026
## V43      0.94      0.94    0.94      0.57  15   0.0026
## V46      0.94      0.94    0.94      0.57  15   0.0026
```

```
## V47      0.94      0.94    0.94      0.58  15   0.0025
## V50      0.94      0.94    0.94      0.58  15   0.0025
## V52      0.94      0.94    0.94      0.58  15   0.0026
##
##  Item statistics
##        n raw.r std.r r.cor r.drop mean  sd
## V28 1278  0.84  0.84  0.83   0.80  3.1 1.3
## V29 1278  0.81  0.81  0.80   0.77  3.0 1.2
## V30 1278  0.80  0.80  0.78   0.75  2.8 1.3
## V31 1278  0.64  0.65  0.60   0.58  3.0 1.2
## V32 1278  0.85  0.85  0.84   0.82  2.7 1.3
## V33 1278  0.80  0.79  0.77   0.75  2.4 1.3
## V34 1278  0.78  0.77  0.75   0.73  2.4 1.4
## V43 1278  0.80  0.80  0.78   0.75  2.3 1.3
## V46 1278  0.81  0.80  0.79   0.76  2.5 1.4
## V47 1278  0.74  0.74  0.71   0.68  3.2 1.3
## V50 1278  0.75  0.75  0.72   0.70  2.9 1.3
## V52 1278  0.78  0.79  0.76   0.74  3.0 1.2
##
## Non missing response frequency for each item
##        0    1    2    3    4    5 miss
## V28 0.02 0.12 0.12 0.33 0.27 0.13    0
## V29 0.02 0.13 0.13 0.39 0.22 0.11    0
## V30 0.03 0.19 0.17 0.28 0.22 0.11    0
## V31 0.02 0.11 0.15 0.35 0.27 0.10    0
## V32 0.02 0.20 0.19 0.30 0.19 0.09    0
## V33 0.03 0.29 0.21 0.25 0.14 0.08    0
## V34 0.03 0.33 0.18 0.19 0.18 0.09    0
## V43 0.03 0.33 0.23 0.23 0.11 0.06    0
## V46 0.02 0.30 0.18 0.26 0.14 0.10    0
## V47 0.02 0.13 0.10 0.27 0.30 0.18    0
## V50 0.02 0.15 0.14 0.34 0.25 0.10    0
## V52 0.02 0.12 0.14 0.41 0.21 0.10    0
```

b)

```r
pca <- prcomp(music)
pca$sdev
```

```
##  [1] 3.5079660 1.1715773 1.0560283 0.9252190 0.8715732 0.8055641 0.7721219
##  [8] 0.7408483 0.7128224 0.6874437 0.6526254 0.5724456
```

3 eigenvalues are greater than 1.

c)

A person's similarity of taste in music with friends or a person's willingless to share his/her personal taste in music.