hw1_solutions

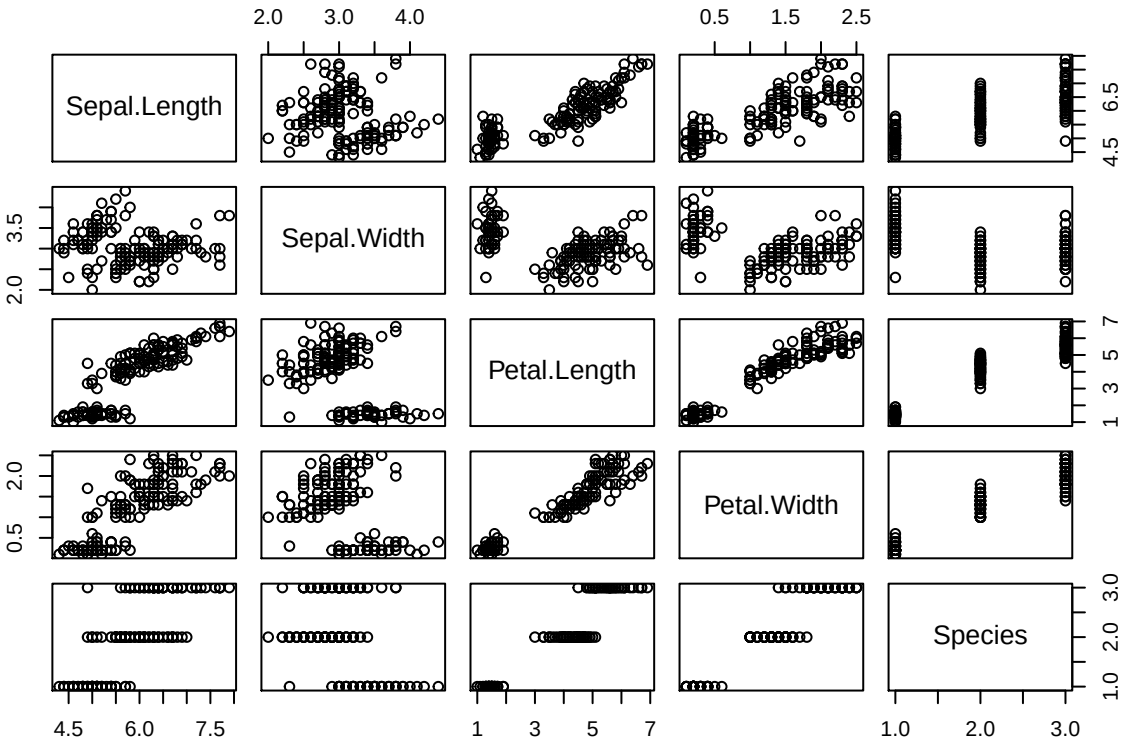
chandan u

2/1/2017

Question 2

From the below pairs plaot, it can be seen that both petal.length and petal.width features have same correlation with other features in the data such as sepal.length and sepal.width. Hence we can remove one of these features as they provide us the similar amount of information(having both of them is redundent.). What we need is un-correlated features which means each features has some or the other differnt information to help us for classification/regresison.

```
data(iris)
pairs(~Sepal.Length + Sepal.Width + Petal.Length + Petal.Width + Species , data=iris)
```



Question 3

a

Yes this is a resonable model As long as the poeple who are measuring are unbiased estimators and also given that the samlpe size is large enough.(and hundred might be good start)

b

Irrespective of the distribution of the population the sample mean distribution is approximately normal (if sample sizes are large enough).

$nbar = \mu$ and $sd = \frac{\sigma}{\sqrt{(n)}}$

c

We have to find the z statistic using the CLT theorem: So the probability that sample mean misses the population mean by two beans is :

P(x< - Z) + P(x > Z) which is nothing but, 1 - (pnorm(z) - pnorm(-z))

where z is Z statistic

```
# n_bar - actualmean
difference = 2
sd = 10
# as per CLT the z statistic is
z = difference/sd

# the probability for more n_bar misssing mean by 2 beans is
1 - (pnorm(z) - pnorm(-z))
```

```
## [1] 0.8414806
```

Hence the probabiliy that the n_bar misses population mean by two beans is 0.8414806.

d

n_bar - n_{true} gives the bias of the sample mean.

Question 4

a

compute the sample covariance matrix:

```
# load data: trees
data("trees")

# column names
#names(trees)


# convert the data frame to matrix: all columns are numeric
# hence can be converted without any hassle
X = as.matrix(trees)

#par(mfrow = c(2, 1))
# pairs plot before scaling
#pairs( X )

# scaling
X = scale(X,center=TRUE,scale=FALSE)

# pairs plot after scaling
# the numeric values of each column has been scaled with mean (subtracted)
```

```
# the correlation still remains the same
#x11()
#pairs(X)

# compute the covariace matrix (X-u)(X-u)^T   (n*1)(1*n) matrix = n*n(matrix)
n = nrow(trees)
cov = t(X) %*% X/n
cov
```

```
##            Girth   Height    Volume
## Girth   9.530239 10.04839  48.27882
## Height 10.048387 39.29032  60.63871
## Volume 48.278824 60.63871 261.48658
```

b

Perform the PCA by computing the SVD of the sample co-variance matrix:

```
# singular value decomposition
svd =  svd(cov) # take the singular value decomposition S = UDU^t
svd
```

```
## $d
## [1] 285.7451321  24.0197024   0.5423039
##
## $u
##             [,1]       [,2]        [,3]
## [1,] -0.1755956  0.0909132 -0.98025557
## [2,] -0.2419510 -0.9691715 -0.04654394
## [3,] -0.9542672  0.2290009  0.19217878
##
## $v
##             [,1]       [,2]        [,3]
## [1,] -0.1755956  0.0909132 -0.98025557
## [2,] -0.2419510 -0.9691715 -0.04654394
## [3,] -0.9542672  0.2290009  0.19217878
```

Now lets perform the PCA:

```
dim(X)
```

```
## [1] 31  3
```

```
dim(svd$u)
```

```
## [1] 3 3
```

```
X %*% svd$u
```

```
##              [,1]         [,2]         [,3]
##  [1,]  21.2828338   0.81468587  1.311169411
##  [2,]  22.4399101   5.68781729  1.249812458
##  [3,]  22.9841197   7.62144283  1.127631353
##  [4,]  14.5915916   0.47325738  0.233809806
##  [5,]  10.0886723  -7.68150126  0.080092264
##  [6,]   8.7283702  -9.40465212  0.061939718
##  [7,]  16.7189135   6.15054222 -0.130797338
```

```
##  [8,]  12.0602598  -1.97659888 -0.050028012
##  [9,]   6.6341696  -5.80576109  0.464813328
## [10,]  10.4028865  -1.56911472  0.080624794
## [11,]   5.3141739  -4.45200553  0.622792199
## [12,]   9.0761224  -2.26820259  0.049426390
## [13,]   8.6944155  -2.17660224  0.126297900
## [14,]  10.4308205   4.61197208  0.138810956
## [15,]  11.0258238  -1.67958487 -0.857322682
## [16,]   8.1515104   0.08131127 -1.097254545
## [17,]  -5.5794501  -7.92316479  0.620035877
## [18,]   0.2156708 -10.32157671 -1.048554459
## [19,]   5.3969516   3.86305943 -1.069201453
## [20,]   7.8364628  10.47315048 -0.995162427
## [21,]  -4.7469354  -0.87865920  0.002087499
## [22,]  -2.5940083  -3.44002204 -0.825152074
## [23,]  -5.5846102   3.45568498  0.044057286
## [24,]  -7.2726361   5.98839956 -0.948880628
## [25,] -12.6384187   2.15451990 -0.649308280
## [26,] -25.9964385   1.29995856  0.644148709
## [27,] -26.5597888   0.41766998  0.459207285
## [28,] -28.6272198   2.98778056  0.659857761
## [29,] -22.1557623   1.43966581 -0.744983471
## [30,] -21.6786287   1.32516537 -0.841072859
## [31,] -48.6397816   0.73136246  1.281103231
```

c

To build a principal component regression model we will use the column of U associated with the smallest value in the Diagonal matrix D:

The smallest element of D is third element.

```
x = svd$u[,3]
lm( x[3] ~ x[1] + x[2])
```

```
##
## Call:
## lm(formula = x[3] ~ x[1] + x[2])
##
## Coefficients:
## (Intercept)         x[1]         x[2]
##      0.1922           NA           NA
```

Question 5

```
X = read.csv("./data/mystery.csv")

X = as.matrix(X)
X = scale(X,center=TRUE,scale=FALSE)
n = nrow(X)
cov = t(X) %*% X/n
svd = svd(cov)
```

```
svd$d
```

```
##  [1] 4.297242e+01 2.925341e+01 1.791203e+01 1.514148e+01 5.476704e+00
##  [6] 1.670427e-14 1.083814e-14 1.005121e-14 9.196282e-15 6.721528e-15
## [11] 6.383136e-15 5.164419e-15 4.666306e-15 3.729708e-15 3.255959e-15
## [16] 2.906601e-15 1.829613e-15 1.672262e-15 9.716415e-16 1.411513e-16
```

As you can see in the diagnoal matrix the first five values in the diagonal are positive and large. Hence if we consider only these five values we wont have any loss of information. Now lets compute the principal components

```
PCA <- X %*% svd$u[,1:5]
PCA[1:5,]
```

```
##            [,1]      [,2]      [,3]        [,4]       [,5]
## [1,] -11.980984  6.245815  5.337389  -1.4271634  1.3461981
## [2,]  -5.889995  5.090747  6.562399   4.2927180 -0.2775179
## [3,]   1.147084 -5.844983 -4.234940 -11.8587524  2.1176356
## [4,]  -6.617753 -3.089898  2.155980   0.1591853 -3.1854632
## [5,]  -6.069610  2.192013 -1.253078  -2.1356202 -2.6170484
```

Question 16

a

If a term occurs in one document : The the tf-idf transformation to data means, the term is more important.

If a term occurs in every document : Then the transformation assigns zero values making the term least important as it occurs everywhere.

b

The $log_(m/df_i)$ only offsets the occurance of a word across the documents. As some words appear more frequently and provide less information as they are common such as (the word the, a etc). Some words are very specific to a document and may give important information related to that document.

Question 17

a

The interval of (a,b) in terms of X is (a^2, b^2)

as a,b is square root values in x^*

b

Since the relation is linear we can consider this as:

$y = m\sqrt(x)$ $y^2 = m^2x$ Which is a parablic equation. Hence y relates to x as a parabola.

Question 25

a

Given: There are three points: x, y and z which belongs to set S and d(x,z∈S)
and d(x,y) can be easily computed once using the distance formula. (This is a one time calculation).

We have to find all the points from y to points in S that are in the range of ϵ. i.e d(y,z∈S) <= ϵ.

The usual way to find these points is to use the distance formula (euclidian distance) to compute the distances from y to z. But our goal is to reduce the calculations . In order acheive that we will use the triangle inequality i.e sum of two sides is always greater than the third side:

so in triange Δxyz: d(x,y) $\leq$ d(y,z) + d(x,z) In the above equation d(x,z) is given And , d(x,y) is just computed once, so using these we can derive the equation as: d(x,y) - d(x,z) $\leq \epsilon$
i.e if the difference is less than equal to ϵ then the point z is within ϵ distance. if the difference is greater than z is not within ϵ distance. We are not doing direct distance calculations. Instead we are using given distances to infer the nature of the third distance.

b

When $x \neq y$ we only calcuate the d(x,y) just once. If x = y then we need not compute d(x,y) as they are same points and d(x,S) is already known.

c

Let x be the point and we have to find all the points that are within β distance of x. Let this point be y. and $x \neq y$.

Now consider another set of points x' , y' such that , x' is epsilon distance from x, i.e and y' is epsilon distance from y (every point has atleast one other point which is in range of epsilon). so D(x,x') and D(y,y') = ϵ

Therefore by triange equalitly:
D(x',y') - D(x,x') - D(y,y') $\leq \beta$ D(x',y') - 2$\epsilon \leq \beta$ But we already have the value of D(x',y') (from the distance matrix.)

Similarly,

D(x',y') + D(x,x') + D(y,y') $\geq \beta$ D(x',y') + 2$\epsilon \geq \beta$

So the domain is :

D(x',y') - 2$\epsilon \leq \beta$ D(x',y') + 2

So when we are using triangle inequalities to find the point within distance β , the above inequalities should be satisfied.

Quesiton 26

d(x,y) = 1 - j(x,y) let, d(x,y) = 0 1-j(x,y) = 0 j(x,y) = 1

$1 = \frac{f_{11}}{/}(f_{11} + f_{01} + f_{10})$ $f_{11} = f_{11} + f_{01} + f_{10}$

where f_{10} means x is 1 and y is 0 So $f_{01} and f_{10}$ should be zero for the above eqution to be true.

But since $f_{01} and f_{10} is 0$, then it means x = y this proves the first axiom

Second Axiom: d(x,y) = d(y,x)

1 - $\frac{f_{11}}{/}(f_{11} + f_{01} + f_{10})$ = d(x,y) here f_{01} means x is zero and y is one similarly,

1 - $\frac{f_{11}}{/}(f_{11} + f_{01} + f_{10})$ = d(y,x) here f_{01} means x is one and y is zero

But still these two equations are same as they same elements in the numerator and the denominator.

So therefore d(x,y) = d(y,x) which proves the axiom tow