STAT 665 Homework 1

Issac Li (zl368)

1/26/2017

Part I

```
myknn <- function(xtrain, xtest, ytrain, k=3) {
  ytrain=as.numeric(as.matrix(ytrain))
  cal_d<-function(xvec1,xmatrix){
    apply(xmatrix,1,function (x) sqrt(sum((xvec1-x)^2)))
  }
  dmatrix=apply(xtest,1,cal_d,xmatrix=xtrain)
  kmatrix=apply(t(dmatrix),1,order)[2:k+1,]
  yresponse=colMeans(apply(kmatrix,2,function(x) ytrain[x]))
return(yresponse)
}
```

Part II

Now read the two datasets and do model selection with training data.

```
# Read two datasets
require(data.table)

## Loading required package: data.table

train=fread("/Users/lizhuo/Documents/STAT665/HW1/citibike_train.csv",header =
TRUE)
weather=fread("/Users/lizhuo/Documents/STAT665/HW1/weather.csv",header =
TRUE)
# Merge two datasets
setkey(train,date)
setkey(weather,date)
mtrain=train[weather,nomatch=0]

# Preprocess the training data
mtrain$holiday=as.numeric(mtrain$holiday)
mtrain$date=as.Date(mtrain$date,format = '%m/%d/%y')
mtrain$day=weekdays(mtrain$date,abbreviate = T)
head(mtrain,10)

##           date trips n_stations holiday PRCP SNWD SNOW TMAX TMIN AWND day
##  1: 2014-01-01  6059        323       1 0.00  0.0    0   33   24  5.6 Wed
##  2: 2015-01-01  5317        327       1 0.00  0.0    0   39   27  7.2 Thu
##  3: 2016-01-01 11009        460       1 0.00  0.0    0   42   34  7.6 Fri
##  4: 2014-01-10  9847        327       0 0.11  0.0    0   37   30  3.4 Fri
##  5: 2015-01-10  6109        328       0 0.00  1.2    0   23   16  8.1 Sat
```

```
##  6: 2016-01-10 14275        466       0 1.80  0.0    0   59   40  9.8 Sun
##  7: 2014-01-11  7695        326       0 0.50  0.0    0   58   37  7.2 Sat
##  8: 2015-01-11  7467        328       0 0.00  1.2    0   37   18  6.0 Sun
##  9: 2016-01-11 22937        471       0 0.00  0.0    0   40   26 10.5 Mon
## 10: 2014-01-12 12515        326       0 0.05  0.0    0   54   38  8.3 Sun
```

We see that the variable `holidays` actually does dicriminate between weekdays and weekends. Thus inferring `weekdays` from `dates` can indeed be beneficial.

```
# Add weekdays since intuitively weekdays have impact on traffic
mtrain$weekday=ifelse(!mtrain$day %in% c("Sat","Sun"),ifelse(mtrain$day
=="Mon",1,

ifelse(mtrain$day == "Tue",2,

ifelse(mtrain$day == "Wed",3,

ifelse(mtrain$day == "Thu",4,5)))),0)
mtrain$weekend=ifelse(mtrain$day %in% c("Sat","Sun"),ifelse(mtrain$day ==
"Sat",1,2),0)
str(mtrain[,!c("date"),with=F])
```

```
## Classes 'data.table' and 'data.frame':   1001 obs. of  12 variables:
##  $ trips     : int  6059 5317 11009 9847 6109 14275 7695 7467 22937 12515
...
##  $ n_stations: int  323 327 460 327 328 466 326 328 471 326 ...
##  $ holiday   : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ PRCP      : num  0 0 0 0.11 0 1.8 0.5 0 0 0.05 ...
##  $ SNWD      : num  0 0 0 0 1.2 0 0 1.2 0 0 ...
##  $ SNOW      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ TMAX      : int  33 39 42 37 23 59 58 37 40 54 ...
##  $ TMIN      : int  24 27 34 30 16 40 37 18 26 38 ...
##  $ AWND      : num  5.6 7.2 7.6 3.4 8.1 9.8 7.2 6 10.5 8.3 ...
##  $ day       : chr  "Wed" "Thu" "Fri" "Fri" ...
##  $ weekday   : num  3 4 5 5 0 0 0 0 1 0 ...
##  $ weekend   : num  0 0 0 0 1 2 1 2 0 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
head(mtrain,5)
```

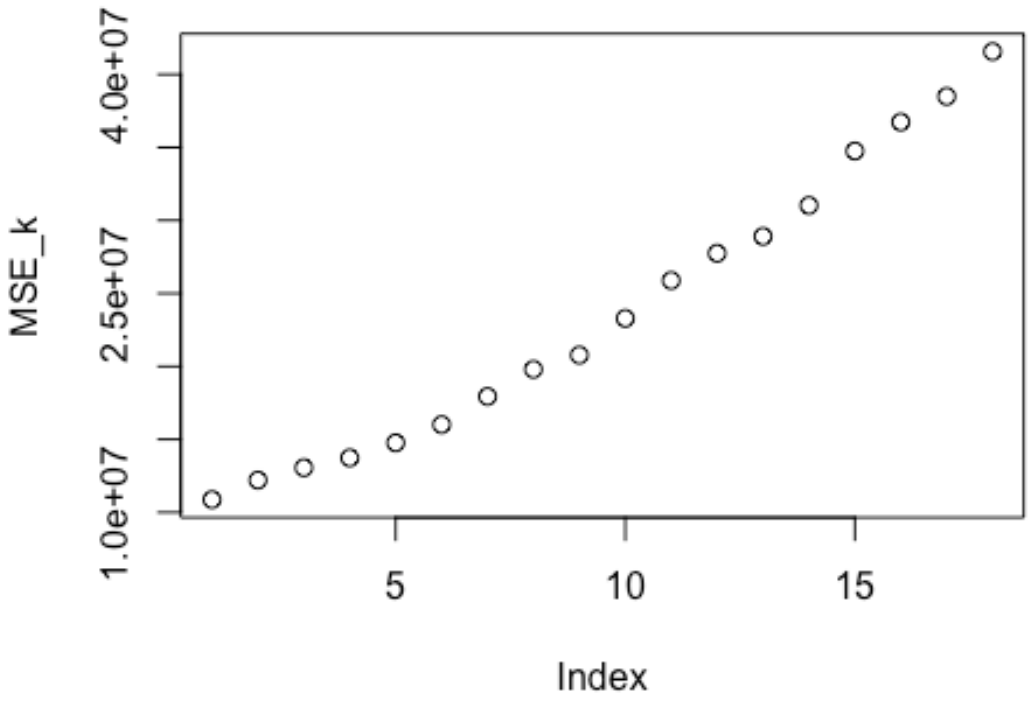
```
##          date trips n_stations holiday PRCP SNWD SNOW TMAX TMIN AWND day
## 1: 2014-01-01  6059        323       1 0.00  0.0    0   33   24  5.6 Wed
## 2: 2015-01-01  5317        327       1 0.00  0.0    0   39   27  7.2 Thu
## 3: 2016-01-01 11009        460       1 0.00  0.0    0   42   34  7.6 Fri
## 4: 2014-01-10  9847        327       0 0.11  0.0    0   37   30  3.4 Fri
## 5: 2015-01-10  6109        328       0 0.00  1.2    0   23   16  8.1 Sat
##    weekday weekend
## 1:       3       0
## 2:       4       0
## 3:       5       0
```

```
## 4:       5       0
## 5:       0       1
```

```
# Standarize the data to [0,1]
mtrain1=mtrain
mtrain=data.table(cbind(mtrain[,c("trips","weekday","weekend"),with=F],

apply(mtrain[,!c("trips","date","day","weekday","weekend"),with=F],
                              2,function(x) scale(x,center =
min(x),scale=max(x)-min(x)))))

head(mtrain)
```

```
##    trips weekday weekend n_stations holiday      PRCP       SNWD SNOW
## 1:  6059       3       0  0.1693989       1 0.0000000 0.00000000    0
## 2:  5317       4       0  0.1912568       1 0.0000000 0.00000000    0
## 3: 11009       5       0  0.9180328       1 0.0000000 0.00000000    0
## 4:  9847       5       0  0.1912568       0 0.0221328 0.00000000    0
## 5:  6109       0       1  0.1967213       0 0.0000000 0.06349206    0
## 6: 14275       0       2  0.9508197       0 0.3621730 0.00000000    0
##          TMAX      TMIN      AWND
## 1: 0.21686747 0.2976190 0.9991112
## 2: 0.28915663 0.3333333 0.9992710
## 3: 0.32530120 0.4166667 0.9993109
## 4: 0.26506024 0.3690476 0.9988915
## 5: 0.09638554 0.2023810 0.9993609
## 6: 0.53012048 0.4880952 0.9995306
```

```
# In-sample train/test split
set.seed(111)
train_ind <- sample(seq_len(nrow(mtrain)), size = floor(nrow(mtrain)*0.75))
xtrain1=mtrain[train_ind,c("trips"),with=FALSE]
ytrain1=mtrain[train_ind,list(trips)]
xtest1=mtrain[-train_ind,c("trips"),with=FALSE]
ytest1=mtrain[-train_ind,list(trips)]

MSE_k=c()
for (k in seq(3,20)){
y_pred=myknn(xtrain = xtrain1,xtest = xtest1,ytrain = ytrain1,k=k)
MSE_k=c(MSE_k,mean(sum((y_pred-ytest1)^2)))
}
plot(MSE_k)
```



Optimal k value selected here is 3. Now let us use linear regression to do the same task.

```
train2=mtrain1[train_ind,!c("date","weekday","weekend"),with=F]
test2=mtrain1[-train_ind]
form1 = trips~.*.-holiday:day
form2 = trips~.
lmfit=lm(formula = form1,data = train2)
bfit=step(lmfit,direction = "backward",trace = 0)
summary(bfit)
```

```
##
## Call:
## lm(formula = trips ~ n_stations + holiday + PRCP + SNWD + TMAX +
##     TMIN + AWND + day + n_stations:holiday + n_stations:PRCP +
##     n_stations:SNWD + n_stations:TMIN + holiday:PRCP + holiday:TMAX +
##     holiday:TMIN + holiday:AWND + PRCP:TMIN + PRCP:AWND + PRCP:day +
##     SNWD:TMIN + TMAX:TMIN + TMAX:AWND + AWND:day, data = train2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14277.1  -2400.1    197.7   2732.6  11941.7
##
## Coefficients:
```

```
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.337e+04  4.976e+03  -2.687 0.007368 **
## n_stations          1.012e+01  1.029e+01   0.983 0.325772
## holiday             6.793e+03  7.428e+03   0.914 0.360797
## PRCP               -1.236e+03  5.043e+03  -0.245 0.806458
## SNWD                3.745e+03  8.336e+02   4.493 8.20e-06 ***
## TMAX                5.806e+02  5.636e+01  10.302  < 2e-16 ***
## TMIN               -4.680e+01  1.023e+02  -0.457 0.647538
## AWND                2.817e+02  3.342e+02   0.843 0.399483
## dayMon              4.170e+02  1.588e+03   0.263 0.792875
## daySat             -8.119e+03  1.264e+03  -6.425 2.41e-10 ***
## daySun             -9.230e+03  1.270e+03  -7.270 9.52e-13 ***
## dayThu              7.426e+02  1.549e+03   0.480 0.631724
## dayTue              6.548e+02  1.534e+03   0.427 0.669675
## dayWed              2.834e+02  1.551e+03   0.183 0.855064
## n_stations:holiday -6.221e+01  1.575e+01  -3.948 8.65e-05 ***
## n_stations:PRCP    -5.737e+01  8.888e+00  -6.454 2.01e-10 ***
## n_stations:SNWD    -1.064e+01  2.568e+00  -4.145 3.80e-05 ***
## n_stations:TMIN     1.292e+00  2.210e-01   5.847 7.63e-09 ***
## holiday:PRCP       -1.254e+04  5.759e+03  -2.178 0.029716 *
## holiday:TMAX        3.089e+02  1.996e+02   1.548 0.122029
## holiday:TMIN       -4.197e+02  2.156e+02  -1.947 0.051898 .
## holiday:AWND        9.195e+02  4.663e+02   1.972 0.049023 *
## PRCP:TMIN           7.867e+01  4.547e+01   1.730 0.084060 .
## PRCP:AWND           5.156e+02  2.317e+02   2.225 0.026389 *
## PRCP:dayMon         6.464e+02  2.593e+03   0.249 0.803184
## PRCP:daySat         2.321e+03  2.372e+03   0.979 0.328043
## PRCP:daySun         9.751e+03  2.555e+03   3.816 0.000147 ***
## PRCP:dayThu         4.784e+03  2.217e+03   2.158 0.031286 *
## PRCP:dayTue         3.858e+03  2.343e+03   1.647 0.099988 .
## PRCP:dayWed         6.174e+03  2.073e+03   2.979 0.002994 **
## SNWD:TMIN          -2.746e+01  6.277e+00  -4.375 1.39e-05 ***
## TMAX:TMIN          -4.434e+00  6.817e-01  -6.505 1.47e-10 ***
## TMAX:AWND          -1.314e+01  4.629e+00  -2.840 0.004642 **
## AWND:dayMon        -1.950e+02  2.738e+02  -0.712 0.476450
## AWND:daySat         5.508e+02  2.046e+02   2.692 0.007267 **
## AWND:daySun         3.628e+02  2.067e+02   1.756 0.079604 .
## AWND:dayThu        -1.096e+02  2.575e+02  -0.426 0.670484
## AWND:dayTue        -1.345e+02  2.618e+02  -0.514 0.607491
## AWND:dayWed         9.177e+01  2.724e+02   0.337 0.736323
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4361 on 711 degrees of freedom
## Multiple R-squared:  0.8628, Adjusted R-squared:  0.8555
## F-statistic: 117.7 on 38 and 711 DF,  p-value: < 2.2e-16
```

We can see there are multiple significant interaction terms.

```
lmfit2=lm(form2,data=train2)
anova(lmfit2,bfit,test = "Chisq")
```

```
## Analysis of Variance Table
##
## Model 1: trips ~ n_stations + holiday + PRCP + SNWD + SNOW + TMAX + TMIN +
##     AWND + day
## Model 2: trips ~ n_stations + holiday + PRCP + SNWD + TMAX + TMIN + AWND +
##     day + n_stations:holiday + n_stations:PRCP + n_stations:SNWD +
##     n_stations:TMIN + holiday:PRCP + holiday:TMAX + holiday:TMIN +
##     holiday:AWND + PRCP:TMIN + PRCP:AWND + PRCP:day + SNWD:TMIN +
##     TMAX:TMIN + TMAX:AWND + AWND:day
##   Res.Df        RSS Df  Sum of Sq  Pr(>Chi)
## 1    735 1.8226e+10
## 2    711 1.3523e+10 24 4702641610 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-value << 0.01 indicates that the addition of these interaction terms is significant. Next we compare the models with MSE.

```
pred_l1=predict(object = lmfit,newdata =  test2[,!c("trips"),with=F])
pred_l2=predict(object = lmfit2,newdata =  test2[,!c("trips"),with=F])
pred_b=predict(object = bfit,newdata =  test2[,!c("trips"),with=F])
MSE_b=mean(sum((pred_b-test2$trips)^2))
MSE_l1=mean(sum((pred_l1-test2$trips)^2))
MSE_l2=mean(sum((pred_l2-test2$trips)^2))
MSE_k=min(MSE_k)
table1=rbind(MSE_b,MSE_l1,MSE_l2,MSE_k)
colnames(table1)<-"Value"
table1
```

```
##               Value
## MSE_b  3.915260e+12
## MSE_l1 3.529356e+12
## MSE_l2 5.826585e+09
## MSE_k  1.087666e+07
```

MSE of the model without any interaction terms, `lmfit2` ("n_stations" "holiday" "PRCP" "SNWD" "SNOW" "TMAX" "TMIN" "AWND" "day"), yields the smallest MSE with validation set. But its MSE is still very large compared to model fit by knn. Thus, we fit training data with the knn model and validate with test data.

```
# Merge two datasets
test=fread("/Users/lizhuo/Documents/STAT665/HW1/citibike_test.csv",header =
TRUE)
setkey(test,date)
mtest=test[weather,nomatch=0]

# Preprocess the test data
mtest$holiday=as.numeric(mtest$holiday)
```

```r
mtest$date=as.Date(mtest$date,format = '%m/%d/%y')
mtest$day=weekdays(mtest$date,abbreviate = T)
# Add weekdays since intuitively weekdays have impact on traffic
mtest$weekday=ifelse(!mtest$day %in% c("Sat","Sun"),ifelse(mtest$day
=="Mon",1,
                                                           ifelse(mtest$day
== "Tue",2,
                                                           ifelse(mtest$day
== "Wed",3,
                                                           ifelse(mtest$day
== "Thu",4,5)))),0)
mtest$weekend=ifelse(mtest$day %in% c("Sat","Sun"),ifelse(mtest$day ==
"Sat",1,2),0)
# standarize
mtest=data.table(cbind(mtest[,c("weekday","weekend"),with=F],

apply(mtest[,!c("date","day","weekday","weekend"),with=F],
                                2,function(x) scale(x,center =
min(x),scale=ifelse(max(x)-min(x)==0,1,max(x)-min(x))))))
xtrain=mtrain[,!c("trips"),with=FALSE]
ytrain=mtrain[,c("trips"),with=FALSE]

pred_final=myknn(xtrain = xtrain,ytrain = ytrain , xtest = mtest,k=3)
output=data.table(cbind(test$date,pred_final))
colnames(output)<-c("date","trips")
head(output)
```

```
##       date   trips
## 1:  4/1/16   33418
## 2: 4/10/16    9559
## 3: 4/11/16 13148.5
## 4: 4/12/16   15208
## 5: 4/13/16   14538
## 6: 4/14/16   16265
```

```r
write.csv(output,"HW1_zl368.csv")
```