```r
1    #Bryson Cook
2    #ISYE6501, Spring 2018
3    #Homework 1
4
5    #Part 2.2.1a: SVM
6
7    install.packages("kknn")
8    install.packages("kernlab")
9
10   library(kknn)
11   library(kernlab)
12
13
14   mydata = as.matrix(read.csv("credit_card_data.csv",header=FALSE))
15
16   model = ksvm(mydata[,1:10], mydata[,11],type="C-svc",
     kernel="vanilladot",C=100,scaled=TRUE)
17   # calculate a1-am
18   a = colSums(model@xmatrix[[1]] * model@coef[[1]])
19   a
20
21   # calculate a0
22   a0 = -model@b
23   a0
24
25   # see what the model predicts
26   pred = predict(model,mydata[,1:10])
27   pred
28
29   # see what fraction of the model's predictions match the actual classification
30   sum(pred == mydata[,11]) / nrow(mydata)
31
32
33   #Part 2.2.1b: Vary C values to view accuracy.  C varies on a logarithmic scale.
34
35   Cvals = seq(-10,10)
36   ans = matrix(, nrow = length(Cvals), ncol = 2)
37   y = 0
38   for (x in Cvals){
39       model = ksvm(mydata[,1:10], mydata[,11],type="C-svc",
         kernel="vanilladot",C=1*10^x,scaled=TRUE)
40       pred = predict(model,mydata[,1:10])
41       accuracy = sum(pred == mydata[,11]) / nrow(mydata)
42       y=y+1
43       ans[y,1]= 1*10^x
44       ans[y,2] = accuracy
45       }
46
47   #Prints C values then the accuracy:
48   plot(log10(ans[,1]), ans[,2], main="KSVM Accuracies",
49         xlab="Log(C Value)", ylab="Accuracy")
50
51   # As C increases, the max accuracy is found between 1*10^-3 and 100, and also at
     100,000.  As C increases,
52   # we are decreasing the margin, but error should also get smaller, though only up to
53   # a certain point (C = 1*10^5 apparently).
54
55
56   ################################################################
57   #Part 3, K-Nearest Neighbor
58
59   mydata = data.frame(read.csv("credit_card_data.csv",header=FALSE))
60   steps = seq(1,100)
61   answery = matrix(, nrow = length(steps), ncol = 2)
62   z=1
63
```

```r
64    for (y in steps){
65       answerx = matrix(, nrow(mydata), ncol = 1)
66       #answerx = matrix(, nrow=100, ncol = 1)
67       #for (x in 1:100){
68       for (x in 1:nrow(mydata)){
69          knn = kknn(V11~., mydata[-x,], mydata[x,], k = y, scale=TRUE)
70          class = round(fitted(knn),0)
71          match = class == mydata[x,11]
72          answerx[x,1] = match
73          total = sum(answerx)
74       }
75       answery[z,1] = y
76       accuracy = total / nrow(mydata)
77       answery[z,2] = accuracy
78       z = z+1
79    }
80    #The max accuracy,0.8532110 occurs at a k-values of 12 and 15
81    plot(answery[,1], answery[,2], main="K-Nearest Neighbor Accuracies",
82         xlab="k Values", ylab="Accuracy")
83
```

```
1    > #Bryson Cook
2    > #ISYE6501, Spring 2018
3    > #Homework 1
4    >
5    > #Part 2.2.1a: SVM
6    >
7    > install.packages("kknn")
8    Installing package into 'C:/Users/212450481/Documents/R/win-library/3.4'
9    (as 'lib' is unspecified)
10   also installing the dependencies 'irlba', 'pkgconfig', 'igraph'
11
12   trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/irlba_2.3.2.zip'
13   Content type 'application/zip' length 279988 bytes (273 KB)
14   downloaded 273 KB
15
16   trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/pkgconfig_2.0.1.zip'
17   Content type 'application/zip' length 20040 bytes (19 KB)
18   downloaded 19 KB
19
20   trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/igraph_1.1.2.zip'
21   Content type 'application/zip' length 8246053 bytes (7.9 MB)
22   downloaded 7.9 MB
23
24   trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/kknn_1.3.1.zip'
25   Content type 'application/zip' length 334837 bytes (326 KB)
26   downloaded 326 KB
27
28   package 'irlba' successfully unpacked and MD5 sums checked
29   package 'pkgconfig' successfully unpacked and MD5 sums checked
30   package 'igraph' successfully unpacked and MD5 sums checked
31   package 'kknn' successfully unpacked and MD5 sums checked
32
33   The downloaded binary packages are in
34       C:\Users\212450481\AppData\Local\Temp\Rtmpq0ayYb\downloaded_packages
35   > install.packages("kernlab")
36   Installing package into 'C:/Users/212450481/Documents/R/win-library/3.4'
37   (as 'lib' is unspecified)
38   trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/kernlab_0.9-25.zip'
39   Content type 'application/zip' length 2218659 bytes (2.1 MB)
40   downloaded 2.1 MB
41
42   package 'kernlab' successfully unpacked and MD5 sums checked
43
44   The downloaded binary packages are in
45       C:\Users\212450481\AppData\Local\Temp\Rtmpq0ayYb\downloaded_packages
46   >
47   > library(kknn)
48   > library(kernlab)
49   >
50   >
51   > mydata = as.matrix(read.csv("credit_card_data.csv",header=FALSE))
52   >
53   > model = ksvm(mydata[,1:10], mydata[,11],type="C-svc",
     kernel="vanilladot",C=100,scaled=TRUE)
54    Setting default kernel parameters
55   > # calculate a1-am
56   > a = colSums(model@xmatrix[[1]] * model@coef[[1]])
57   > a
58             V1              V2              V3              V4              V5              V6
59   -0.0010065348 -0.0011729048 -0.0016261967  0.0030064203  1.0049405641 -0.0028259432
60             V7              V8              V9             V10
61    0.0002600295 -0.0005349551 -0.0012283758  0.1063633995
62   >
63   > # calculate a0
64   > a0 = -model@b
65   > a0
```
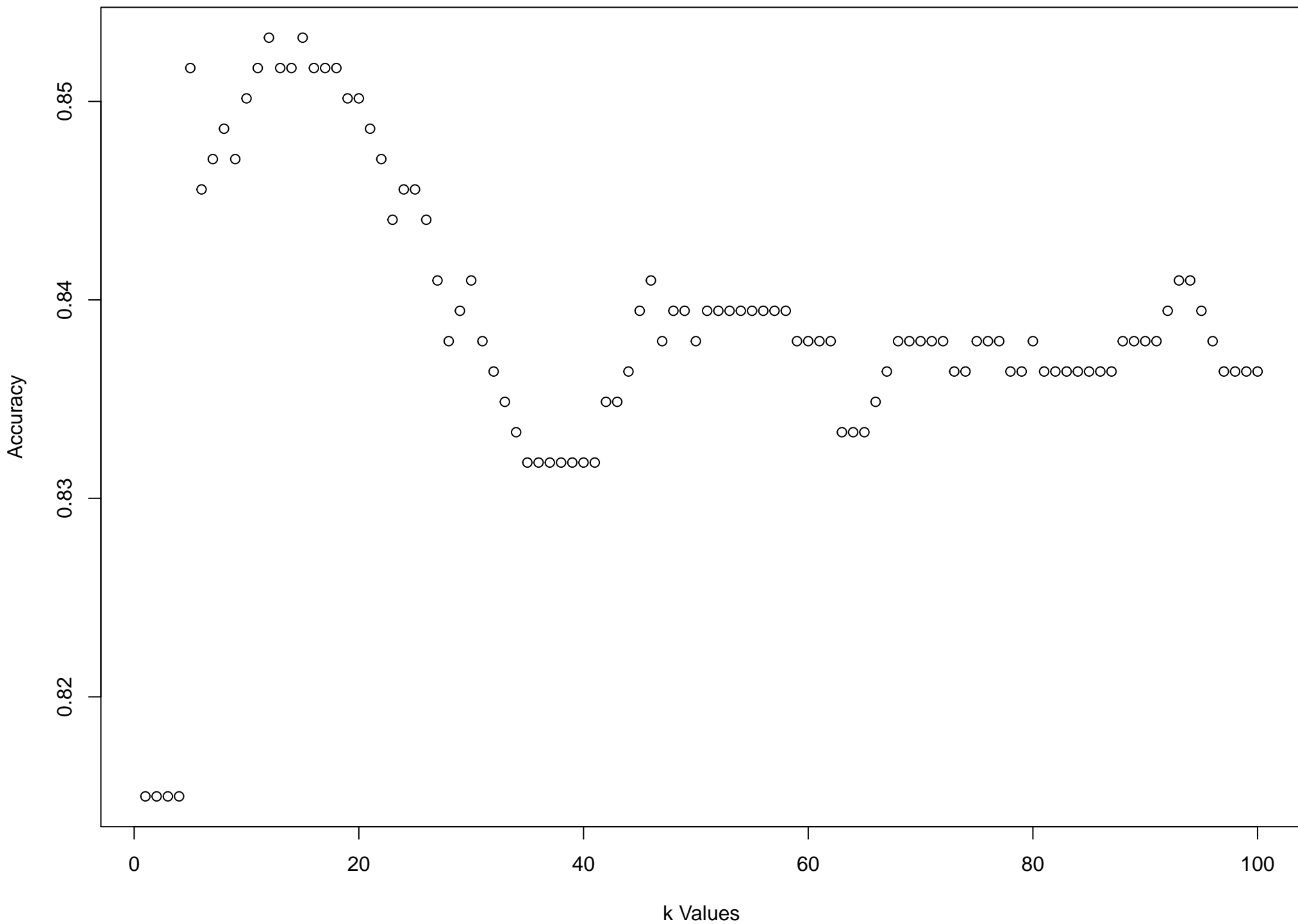
```
66   [1] 0.08158492
67   >
68   > # see what the model predicts
69   > pred = predict(model,mydata[,1:10])
70   > pred
71     [1] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
       1 1
72    [44] 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
       1 1
73    [87] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
       1 1
74   [130] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
     1 1
75   [173] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     1 1
76   [216] 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
     0 0
77   [259] 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
     0 0
78   [302] 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
     0 0
79   [345] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0
80   [388] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0
81   [431] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
     1 1
82   [474] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     1 1
83   [517] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
     1 1
84   [560] 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0
85   [603] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0
86   [646] 0 0 0 0 0 0 0 0 0 0
87   >
88   > # see what fraction of the model's predictions match the actual classification
89   > sum(pred == mydata[,11]) / nrow(mydata)
90   [1] 0.8639144
91   >
92   >
93   > #Part 2.2.1b: Vary C values to view accuracy.  C varies on a logarithmic scale.
94   >
95   > Cvals = seq(-10,10)
96   > ans = matrix(, nrow = length(Cvals), ncol = 2)
97   > y = 0
98   > for (x in Cvals){
99   +    model = ksvm(mydata[,1:10], mydata[,11],type="C-svc",
     kernel="vanilladot",C=1*10^x,scaled=TRUE)
100  +    pred = predict(model,mydata[,1:10])
101  +    accuracy = sum(pred == mydata[,11]) / nrow(mydata)
102  +    y=y+1
103  +    ans[y,1]= 1*10^x
104  +    ans[y,2] = accuracy
105  +    }
106   Setting default kernel parameters
107   Setting default kernel parameters
108   Setting default kernel parameters
109   Setting default kernel parameters
110   Setting default kernel parameters
111   Setting default kernel parameters
112   Setting default kernel parameters
113   Setting default kernel parameters
114   Setting default kernel parameters
115   Setting default kernel parameters
```

```
116    Setting default kernel parameters
117    Setting default kernel parameters
118    Setting default kernel parameters
119    Setting default kernel parameters
120    Setting default kernel parameters
121    Setting default kernel parameters
122    Setting default kernel parameters
123    Setting default kernel parameters
124    Setting default kernel parameters
125    Setting default kernel parameters
126    Setting default kernel parameters
127    >
128    > #Prints C values then the accuracy:
129    > plot(log10(ans[,1]), ans[,2], main="KSVM Accuracies",
130    +       xlab="Log(C Value)", ylab="Accuracy")
131    >
132    > # As C increases, the max accuracy is found between 1*10^-3 and 100, and also at
       100,000.  As C increases,
133    > # we are decreasing the margin, but error should also get smaller, though only up to
134    > # a certain point (C = 1*10^5 apparently).
135    >
136    >
137    > ###########################################################################
138    > #Part 3, K-Nearest Neighbor
139    >
140    > mydata = data.frame(read.csv("credit_card_data.csv",header=FALSE))
141    > steps = seq(1,100)
142    > answery = matrix(, nrow = length(steps), ncol = 2)
143    > z=1
144    >
145    > for (y in steps){
146    +    answerx = matrix(, nrow(mydata), ncol = 1)
147    +    #answerx = matrix(, nrow=100, ncol = 1)
148    +    #for (x in 1:100){
149    +    for (x in 1:nrow(mydata)){
150    +      knn = kknn(V11~., mydata[-x,], mydata[x,], k = y, scale=TRUE)
151    +      class = round(fitted(knn),0)
152    +      match = class == mydata[x,11]
153    +      answerx[x,1] = match
154    +      total = sum(answerx)
155    +    }
156    +    answery[z,1] = y
157    +    accuracy = total / nrow(mydata)
158    +    answery[z,2] = accuracy
159    +    z = z+1
160    + }
161    > #The max accuracy,0.8532110 occurs at a k-values of 12 and 15
162    > plot(answery[,1], answery[,2], main="K-Nearest Neighbor Accuracies",
163    +       xlab="k Values", ylab="Accuracy")
```

K−Nearest Neighbor Accuracies

KSVM Accuracies