kNN Clustering - Regression

Using 10-fold cross validations

Karen Mazidi

Load the data

```
library(ISLR)
df <- Auto[]
df$origin <- as.integer(df$origin)
# subset to columns mpg, weight, year, origin
df <- data.frame(scale(df[, c(1, 5, 7, 8)]  ))
```

Create the 10 folds

We could do this manually but there is a function in caret that does this. Since the Auto data is a little less than 400 rows, we expect each of the 10 folds to be of legth 40 or less. We confirm that with sapply.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.3
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
set.seed(1234)
folds <- createFolds(df$mpg, k=10)
sapply(folds, length)
```

```
## Fold01 Fold02 Fold03 Fold04 Fold05 Fold06 Fold07 Fold08 Fold09 Fold10
##     39     41     38     39     38     39     39     39     40     40
```

Look at the fold indices

To get a better idea of the folds, let's just print the indices for each fold.

```
for (i in 1:10){
  print(folds[[i]])
}
```

```
##  [1]   9  12  19  23  25  43  45  53  63  70  92 104 107 108 120 124 134
## [18] 146 161 170 181 185 190 215 216 222 238 245 257 261 313 315 320 321
## [35] 322 332 346 357 365
##  [1]  18  20  22  29  44  67  77  78  89 103 105 112 116 137 141 149 162
## [18] 188 202 227 229 232 259 260 266 272 273 279 292 309 314 325 327 335
## [35] 350 354 364 374 385 388 392
##  [1]  13  15  24  42  54  55  57  60  73  76  84  91 113 126 131 138 143
## [18] 144 157 159 164 173 191 197 204 206 207 218 225 230 270 278 298 310
## [35] 339 351 362 366
##  [1]   1   6  11  32  33  34  85  94  99 101 110 111 114 127 135 147 150
## [18] 153 183 194 196 210 223 250 265 268 282 290 293 294 306 316 317 319
## [35] 359 368 379 380 387
##  [1]  21  35  37  47  58  59  69  88 115 128 132 151 152 156 167 175 184
## [18] 186 187 217 221 248 254 269 276 286 299 305 311 323 329 333 338 342
```

```
## [35] 349 384 386 391
##  [1]  36  38  50  62  66  79  83  87  96 109 142 158 169 178 179 189 201
## [18] 208 231 234 242 246 253 264 284 288 291 297 300 304 308 328 330 334
## [35] 341 356 361 375 383
##  [1]  10  26  27  30  51  64  68  74  95 123 139 148 155 163 166 168 171
## [18] 177 180 182 199 213 220 251 252 277 283 285 302 307 312 324 336 337
## [35] 347 353 372 381 390
##  [1]   8  14  31  39  48  52  72  82  90  93  98 100 102 130 160 165 176
## [18] 192 195 203 209 224 233 239 240 243 249 258 267 275 281 287 289 295
## [35] 303 355 371 378 389
##  [1]   3   4  28  61  75  80  97 106 118 119 121 122 129 133 136 140 172
## [18] 174 193 198 205 212 214 219 237 244 255 262 274 280 296 318 326 331
## [35] 340 352 358 369 373 376
##  [1]   2   5   7  16  17  40  41  46  49  56  65  71  81  86 117 125 145
## [18] 154 200 211 226 228 235 236 241 247 256 263 271 301 343 344 345 348
## [35] 360 363 367 370 377 382
```

Perform 10-fold cv

For now we will just let k=3 and perform 10-fold cv, then average the correlation and mse values.

```
test_mse <- rep(0, 10)
test_cor <- rep(0, 10)
for (i in 1:10){
  fit <- knnreg(df[-folds[[i]], 2:4], df$mpg[-folds[[i]]], k=3)
  pred <- predict(fit, df[folds[[i]], 2:4])
  test_cor[i] <- cor(pred, df$mpg[folds[[i]]])
  test_mse[i] <- mean((pred - df$mpg[folds[[i]]])^2)
}
print(paste("Average correlation is ", round(mean(test_cor), 2)))
```

```
## [1] "Average correlation is  0.93"
```

```
print(paste("range is ", range(test_cor)))
```

```
## [1] "range is  0.90883537818507"  "range is  0.946753085315825"
```

```
print(paste("Average mse is ", round(mean(test_mse), 2)))
```

```
## [1] "Average mse is  0.15"
```

```
print(paste("range is ", range(test_mse)))
```

```
## [1] "range is  0.111756182643287" "range is  0.201818162667268"
```

Try with various k

We modify the code above to be an anonymouse function called by sapply.

```
# try various values for k
k_values <- seq(1, 39, 2)
results <- sapply(k_values, function(k){
  mse_k <- rep(0, 10)
  cor_k <- rep(0, 10)
  for (i in 1:10){
    fit <- knnreg(df[-folds[[i]], 2:4], df$mpg[-folds[[i]]], k=k)
```

```
    pred <- predict(fit, df[folds[[i]], 2:4])
    cor_k[i] <- cor(pred, df$mpg[folds[[i]]])
    mse_k[i] <- mean((pred - df$mpg[folds[[i]]])^2)
  }
  #print(paste(mean(cor_k), mean(mse_k)))
  list(mean(cor_k), mean(mse_k))
})
# reshape results into matrix
m <- matrix(results, nrow=20, ncol=2, byrow=TRUE)
```

Examine results

Plot the correlation and mse for each value of k.

```
par(mfrow=c(2, 1))
plot(1:20, unlist(m[,1]), lwd=2, type="o", col='red', ylab="Correlation")
plot(1:20, unlist(m[,2]), lwd=2, type="o", col='blue', ylab="MSE")
```