Applied Data Mining Homework 2

Xun Zhao, xz2827

1. Classifier g_m

```
train_g = function(m, data){
    vectors = data[-ncol(data)]
    labels = ifelse(data[,ncol(data)] == data[1,ncol(data)], 1, -1)

    pairs = sapply(1:m, function(i){
        x.pos = vectors[sample(row.names(vectors[labels == 1,]), 1),]
        x.neg = vectors[sample(row.names(vectors[labels == -1,]), 1),]

        w = (x.pos - x.neg) / sum((x.pos - x.neg) ^ 2)
        c.w = sum(w * (x.pos + x.neg) / 2)

        misc = sum(sign(as.matrix(w) %*% t(as.matrix(vectors)) - c.w) != labels)
        s = ifelse(misc < nrow(data) / 2, 1, -1)

        v = s * w
        c = sum(v * (x.pos + x.neg) / 2)

        return(c(as.matrix(v), c))
    })
    return(list(t(pairs[1:ncol(vectors),]), as.matrix(pairs[nrow(pairs),])))
}
```

Classifier Function

`V` and `c` is the result of function g_m.

```
classify = function(x, V, c){
    return(sign(sum(sign(V %*% as.matrix(x) - c))))
}
```

Load Data from `wdbcdata.zip`

Shuffle the data and separate it into two parts with equal size.

```
data.d = read.csv('../Data/wdbc.data')
data.l = read.csv('../Data/wdbc.labels')
data = data.frame(data.d, data.l)
head(data)
```

```
##   X17.99 X10.38 X122.8  X1001 X0.1184 X0.2776 X0.3001 X0.1471 X0.2419
## 1  20.57  17.77 132.90 1326.0 0.08474 0.07864  0.0869 0.07017  0.1812
## 2  19.69  21.25 130.00 1203.0 0.10960 0.15990  0.1974 0.12790  0.2069
## 3  11.42  20.38  77.58  386.1 0.14250 0.28390  0.2414 0.10520  0.2597
## 4  20.29  14.34 135.10 1297.0 0.10030 0.13280  0.1980 0.10430  0.1809
## 5  12.45  15.70  82.57  477.1 0.12780 0.17000  0.1578 0.08089  0.2087
## 6  18.25  19.98 119.60 1040.0 0.09463 0.10900  0.1127 0.07400  0.1794
##   X0.07871 X1.095 X0.9053 X8.589 X153.4 X0.006399 X0.04904 X0.05373
## 1  0.05667 0.5435  0.7339  3.398  74.08  0.005225  0.01308  0.01860
## 2  0.05999 0.7456  0.7869  4.585  94.03  0.006150  0.04006  0.03832
## 3  0.09744 0.4956  1.1560  3.445  27.23  0.009110  0.07458  0.05661
## 4  0.05883 0.7572  0.7813  5.438  94.44  0.011490  0.02461  0.05688
## 5  0.07613 0.3345  0.8902  2.217  27.19  0.007510  0.03345  0.03672
## 6  0.05742 0.4467  0.7732  3.180  53.91  0.004314  0.01382  0.02254
##   X0.01587 X0.03003 X0.006193 X25.38 X17.33 X184.6  X2019 X0.1622 X0.6656
## 1  0.01340  0.01389  0.003532  24.99  23.41 158.80 1956.0  0.1238  0.1866
## 2  0.02058  0.02250  0.004571  23.57  25.53 152.50 1709.0  0.1444  0.4245
## 3  0.01867  0.05963  0.009208  14.91  26.50  98.87  567.7  0.2098  0.8663
## 4  0.01885  0.01756  0.005115  22.54  16.67 152.20 1575.0  0.1374  0.2050
## 5  0.01137  0.02165  0.005082  15.47  23.75 103.40  741.6  0.1791  0.5249
## 6  0.01039  0.01369  0.002179  22.88  27.66 153.20 1606.0  0.1442  0.2576
##   X0.7119 X0.2654 X0.4601 X0.1189 X1
## 1  0.2416  0.1860  0.2750 0.08902  1
## 2  0.4504  0.2430  0.3613 0.08758  1
## 3  0.6869  0.2575  0.6638 0.17300  1
## 4  0.4000  0.1625  0.2364 0.07678  1
## 5  0.5355  0.1741  0.3985 0.12440  1
## 6  0.3784  0.1932  0.3063 0.08368  1
```

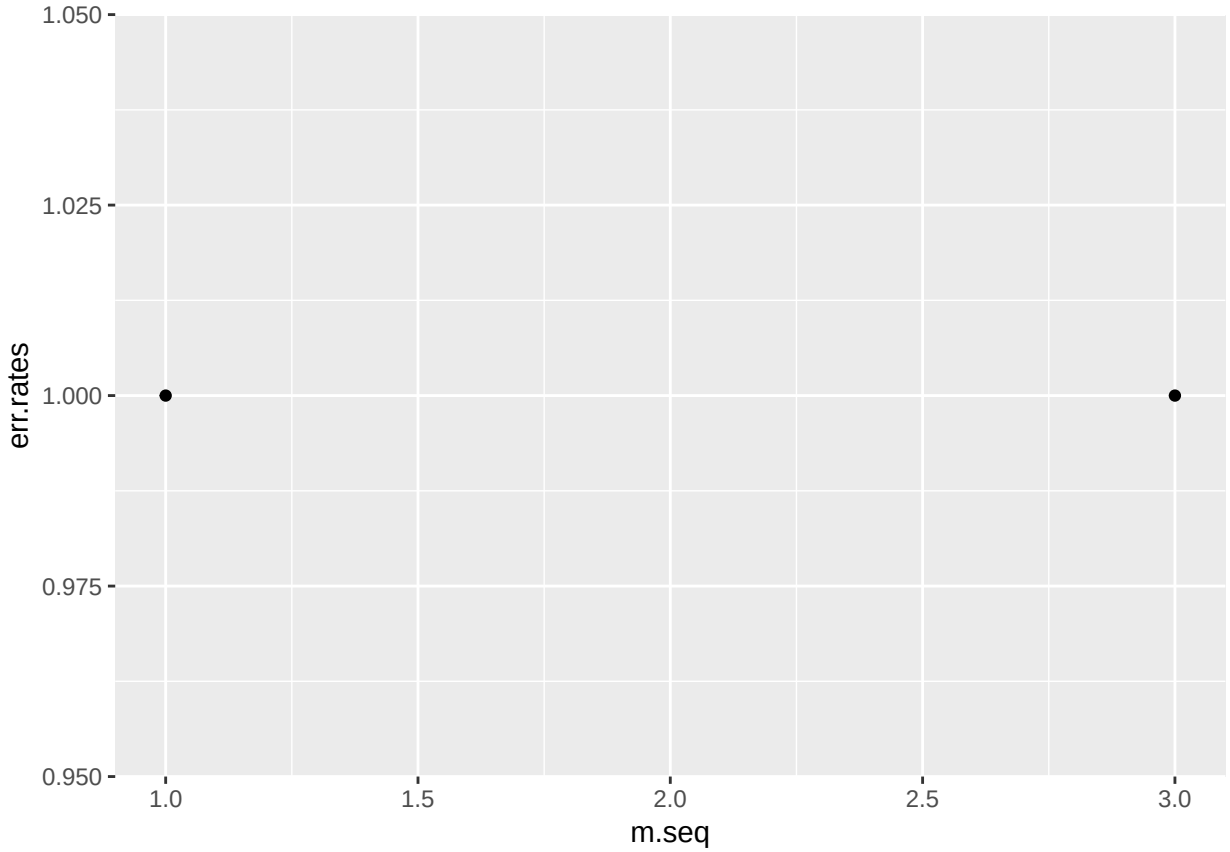
```
data = sample(data)
as.matrix(data[2,])
```

```
##   X0.006399 X0.2654 X1001 X0.006193 X10.38 X2019 X0.7119 X0.01587 X122.8
## 2   0.00615   0.243  1203  0.004571  21.25  1709  0.4504  0.02058    130
##   X0.1189 X0.03003 X153.4 X184.6 X17.33 X0.2419 X0.07871 X0.1471 X1.095
## 2 0.08758   0.0225  94.03  152.5  25.53  0.2069  0.05999  0.1279 0.7456
##   X8.589 X17.99 X0.6656 X25.38 X0.1622 X0.2776 X0.1184 X0.3001 X1 X0.04904
## 2  4.585  19.69  0.4245  23.57  0.1444  0.1599  0.1096  0.1974  1  0.04006
##   X0.05373 X0.4601 X0.9053
## 2  0.03832  0.3613  0.7869
```

```
data.tr = data[1:round(nrow(data) / 2),]
data.ts = data[(round(nrow(data) / 2) + 1):nrow(data),]
```

Train with Different m

```
m.seq = seq(1, 3, 2)
err.rates = sapply(
    m.seq,
    function(m){
        Vc = train_g(m, data.tr)
        V = Vc[[1]]
        c = Vc[[2]]
        esti = apply(
            t(as.matrix(data.ts[,-ncol(data.ts)])), 2,
            classify, V = V, c = c
            )
        err.rate = sum(esti != data.ts[,ncol(data.ts)]) / nrow(data.ts)
        return(err.rate)
        }
    )
library(ggplot2)
graph = ggplot(
    data.frame(cbind(m.seq, err.rates)),
    aes(x = m.seq, y = err.rates)
    ) +
    geom_point()
plot(graph)
```


1.050
1.025
1.000
0.975
0.950
err.rates
1.0
1.5
2.0
2.5
3.0
m.seq