Naive Bayes with the Breast Cancer data

Karen Mazidi

In this notebook we compare Naive Bayes and logistic regression on the breast cancer data in package mlbench.

Load the data

The breast cancer data is in the mlbench package. There are 669 observations with 11 columns. Column 1 is an ID that will be ignored later, columns 2-10 are factors specifying information gleaned from bioposies. The final column is the label: benign or malignant. The class distribution is 458 benign to 241 malignant, about 64% benign to 36% malignant.

```
library(mlbench)
data(BreastCancer)
str(BreastCancer)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ Id             : chr  "1000025" "1002945" "1015425" "1016277" ...
##  $ Cl.thickness   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4 8 1 2 2 4 ...
##  $ Cell.size      : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 1 1 2 ...
##  $ Cell.shape     : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 2 1 1 ...
##  $ Marg.adhesion  : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3 8 1 1 1 1 ...
##  $ Epith.c.size   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare.nuclei    : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1 ...
##  $ Bl.cromatin    : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses        : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 5 1 ...
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
```

```
summary(BreastCancer$Class)
```

```
##    benign malignant
##       458       241
```

Divide data into train, test

First remove the Id column, then divide into 80% train, 20% test.

```
set.seed(1234)
df <- BreastCancer[,-1] # remove ID
i <- sample(1:nrow(df), 0.8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

logistic regression

Build a logistic regression model.

```
glm1 <- glm(Class~., data=train, family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm1)
```

```
## 
## Call:
## glm(formula = Class ~ ., family = binomial, data = train)
## 
## Deviance Residuals: 
##        Min          1Q      Median          3Q         Max  
## -4.232e-05  -2.100e-08  -2.100e-08   2.100e-08   5.038e-05  
## 
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.514e+01  1.515e+05   0.000    1.000
## Cl.thickness.L    4.139e+01  1.828e+05   0.000    1.000
## Cl.thickness.Q    2.477e+01  9.563e+04   0.000    1.000
## Cl.thickness.C   -2.231e-01  1.278e+05   0.000    1.000
## Cl.thickness^4   -4.102e+01  1.331e+05   0.000    1.000
## Cl.thickness^5    7.805e+00  1.545e+05   0.000    1.000
## Cl.thickness^6    3.685e+01  1.539e+05   0.000    1.000
## Cl.thickness^7    6.812e+00  1.312e+05   0.000    1.000
## Cl.thickness^8   -4.545e+01  7.591e+04  -0.001    1.000
## Cl.thickness^9   -4.518e+01  1.037e+05   0.000    1.000
## Cell.size.L       1.097e+01  2.194e+05   0.000    1.000
## Cell.size.Q       4.087e+01  2.150e+05   0.000    1.000
## Cell.size.C       5.496e+00  1.571e+05   0.000    1.000
## Cell.size^4      -3.870e+01  2.079e+05   0.000    1.000
## Cell.size^5      -2.113e+01  1.008e+05   0.000    1.000
## Cell.size^6      -4.133e+01  2.307e+05   0.000    1.000
## Cell.size^7      -1.931e+01  1.206e+05   0.000    1.000
## Cell.size^8       2.570e+01  1.695e+05   0.000    1.000
## Cell.size^9       8.810e+00  2.190e+05   0.000    1.000
## Cell.shape.L      6.360e+01  2.526e+05   0.000    1.000
## Cell.shape.Q     -3.578e+01  1.466e+05   0.000    1.000
## Cell.shape.C     -4.985e+00  1.793e+05   0.000    1.000
## Cell.shape^4      2.534e+01  1.700e+05   0.000    1.000
## Cell.shape^5     -3.471e+01  2.097e+05   0.000    1.000
## Cell.shape^6      2.220e+01  1.784e+05   0.000    1.000
## Cell.shape^7      2.068e+01  1.418e+05   0.000    1.000
## Cell.shape^8      6.037e+01  1.264e+05   0.000    1.000
## Cell.shape^9      2.375e+00  1.324e+05   0.000    1.000
## Marg.adhesion.L   6.802e+01  1.933e+05   0.000    1.000
## Marg.adhesion.Q  -1.248e+01  1.909e+05   0.000    1.000
## Marg.adhesion.C  -2.669e+01  1.993e+05   0.000    1.000
## Marg.adhesion^4  -7.226e+00  2.009e+05   0.000    1.000
## Marg.adhesion^5   2.564e+01  3.074e+05   0.000    1.000
## Marg.adhesion^6   2.167e+01  2.558e+05   0.000    1.000
## Marg.adhesion^7  -2.507e+01  2.279e+05   0.000    1.000
## Marg.adhesion^8  -1.661e+01  2.617e+05   0.000    1.000
## Marg.adhesion^9   8.448e+00  2.206e+05   0.000    1.000
## Epith.c.size.L   -4.147e+01  4.843e+05   0.000    1.000
## Epith.c.size.Q   -3.595e+00  3.726e+05   0.000    1.000
## Epith.c.size.C    2.454e+01  1.706e+05   0.000    1.000
## Epith.c.size^4    6.592e+01  3.006e+05   0.000    1.000
## Epith.c.size^5     3.269e+01  4.152e+05   0.000    1.000
## Epith.c.size^6     2.110e+01  3.335e+05   0.000    1.000
## Epith.c.size^7     5.731e+01  1.724e+05   0.000    1.000
## Epith.c.size^8     2.699e+01  1.353e+05   0.000    1.000
## Epith.c.size^9    -4.999e+00  1.186e+05   0.000    1.000
## Bare.nuclei2       1.307e+01  1.012e+05   0.000    1.000
## Bare.nuclei3       3.042e+01  4.511e+04   0.001    0.999
## Bare.nuclei4       3.640e+01  1.192e+05   0.000    1.000
## Bare.nuclei5       4.132e+01  2.611e+04   0.002    0.999
## Bare.nuclei6       5.184e+01  3.611e+05   0.000    1.000
## Bare.nuclei7       7.735e+01  4.943e+05   0.000    1.000
## Bare.nuclei8       4.455e+01  1.415e+05   0.000    1.000
## Bare.nuclei9       1.290e+02  5.197e+05   0.000    1.000
## Bare.nuclei10      5.082e+01  7.020e+04   0.001    0.999
## Bl.cromatin2       1.417e+01  1.124e+05   0.000    1.000
## Bl.cromatin3       1.066e+01  5.170e+04   0.000    1.000
## Bl.cromatin4       8.994e+00  1.274e+05   0.000    1.000
## Bl.cromatin5      -4.868e-01  7.020e+04   0.000    1.000
## Bl.cromatin6       3.660e+01  2.673e+05   0.000    1.000
## Bl.cromatin7       4.728e+00  1.544e+05   0.000    1.000
## Bl.cromatin8       1.746e+01  2.210e+05   0.000    1.000
## Bl.cromatin9      -1.918e+01  2.203e+05   0.000    1.000
## Bl.cromatin10     -2.536e+01  3.244e+05   0.000    1.000
## Normal.nucleoli2  -3.369e+01  2.094e+05   0.000    1.000
## Normal.nucleoli3  -3.208e-01  1.599e+05   0.000    1.000
## Normal.nucleoli4   1.397e+01  1.814e+05   0.000    1.000
## Normal.nucleoli5  -1.425e+01  1.816e+05   0.000    1.000
## Normal.nucleoli6   2.031e+01  1.963e+05   0.000    1.000
## Normal.nucleoli7  -3.473e+01  1.573e+05   0.000    1.000
## Normal.nucleoli8  -2.679e+01  4.728e+05   0.000    1.000
## Normal.nucleoli9   7.097e+01  2.071e+05   0.000    1.000
## Normal.nucleoli10  1.862e+01  2.154e+05   0.000    1.000
## Mitoses2           3.765e+01  1.112e+05   0.000    1.000
## Mitoses3           5.382e+01  1.491e+05   0.000    1.000
## Mitoses4          -1.267e+01  3.089e+05   0.000    1.000
## Mitoses5          -4.642e+01  3.149e+05   0.000    1.000
## Mitoses6          -1.207e+02  4.714e+05   0.000    1.000
## Mitoses7           1.618e+01  1.458e+05   0.000    1.000
## Mitoses8          -1.144e+01  3.349e+05   0.000    1.000
## Mitoses10          1.355e+02  3.444e+05   0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6.9750e+02  on 548  degrees of freedom
## Residual deviance: 2.2578e-08  on 468  degrees of freedom
##   (10 observations deleted due to missingness)
## AIC: 162
##
## Number of Fisher Scoring iterations: 25
```

Test

Evaluate on the test data. The logistic regression model gets 92% accuracy.

```
probs1 <- predict(glm1, newdata=test, type="response")
pred1 <- ifelse(probs1>0.5, 2, 1)
table(pred1, test$Class)
```

```
##
## pred1 benign malignant
##     1     73         7
##     2      4        50
```

```
acc1 <- mean(pred1==as.integer(test$Class), na.rm=TRUE)
acc1
```

```
## [1] 0.9179104
```

Examine the results using the caret package.

```
library(caret)
```

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

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

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

```
confusionMatrix(pred1, as.integer(test$Class), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 73  7
##          2  4 50
##
##                Accuracy : 0.9179
##                  95% CI : (0.8579, 0.9583)
##     No Information Rate : 0.5746
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8309
##  Mcnemar's Test P-Value : 0.5465
##
##             Sensitivity : 0.9481
##             Specificity : 0.8772
##          Pos Pred Value : 0.9125
##          Neg Pred Value : 0.9259
##              Prevalence : 0.5746
##          Detection Rate : 0.5448
##    Detection Prevalence : 0.5970
##       Balanced Accuracy : 0.9126
##
##        'Positive' Class : 1
##
```

Build a Naive Bayes classifier

Use the same test and train data for comparison.

```
library(e1071)
nb1 <- naiveBayes(train[,-10], train[,10])
summary(nb1)
```

```
##         Length Class  Mode     
## apriori 2      table  numeric  
## tables  9      -none- list     
## levels  2      -none- character
## call    3      -none- call     
```

Evaluate on the test data

The Naive Bayes model gets 96% accuracy.

```
pred2 <- predict(nb1, newdata=test[,-10], type="class")
table(pred2, test$Class)
```

```
##            
## pred2       benign malignant
##   benign        78         1
##   malignant      4        57
```

```
acc2 <- mean(pred2==test$Class)
acc2
```

```
## [1] 0.9642857
```

Evaluate the results with the caret package.

```
confusionMatrix(pred2, test$Class, positive="benign")
```

```
## Confusion Matrix and Statistics
## 
##            Reference
## Prediction  benign malignant
##   benign        78         1
##   malignant      4        57
##                                           
##                Accuracy : 0.9643          
##                  95% CI : (0.9186, 0.9883)
##     No Information Rate : 0.5857          
##     P-Value [Acc > NIR] : <2e-16          
##                                           
##                   Kappa : 0.927           
##  Mcnemar's Test P-Value : 0.3711          
##                                           
##             Sensitivity : 0.9512          
##             Specificity : 0.9828          
##          Pos Pred Value : 0.9873          
##          Neg Pred Value : 0.9344          
##              Prevalence : 0.5857          
##          Detection Rate : 0.5571          
##    Detection Prevalence : 0.5643          
##       Balanced Accuracy : 0.9670          
##                                           
##        'Positive' Class : benign          
```

##