Math 5365

Data Mining 1

Homework 9

Mary Barker

1. Verify that for $\beta = 0, 1, \infty$, F_β is equal to p, F_1, r, respectively.

   The formula for F_β is given by

$$F_\beta = \frac{(\beta^2 + 1)rp}{r + \beta^2 p}$$

   F_0, F_1, F_∞ are defined as follows.

$$F_0 = p, \quad F_1 = \frac{2rp}{r+p}, \quad F_\infty = r$$

   For $\beta = 0$,

$$F_\beta = \frac{(0+1)rp}{r+0p} = \frac{rp}{r} = p = F_0$$

   For $\beta = 1$,

$$F_\beta = \frac{(1+1)rp}{r+1p} = \frac{2rp}{r+p} = F_1$$

   For $\beta = \infty$,

$$F_\beta = \lim_{x\to\infty} \frac{(x^2+1)rp}{r+x^2 p} = \lim_{x\to\infty} \frac{rp + rp/x^2}{r/x^2 + p} = \frac{rp}{p} = r = F_\infty$$

2. Find weights $w_i, i = 1, ..., 4$, such that the weighted accuracy is equal to the given performance metric.

   The weighted accuracy is given by

$$\frac{w_1 TP + w_4 TN}{w_1 TP + w_2 FP + w_3 FN + w_4 TN}$$

(a) Accuracy

$P(\hat{Y} = Y) = \frac{TP+TN}{TP+FP+FN+TN}$, so $w_i = 1, i = 1, ..., 4$.

(b) Sensitivity

$P(\hat{Y} = +|Y = +) = \frac{TP}{TP+FN}$, so $w_1 = w_3 = 1$, $w_2 = w_4 = 0$.

(c) Specificity

$P(\hat{Y} = -|Y = -) = \frac{TN}{TN+FP}$, so $w_1 = w_3 = 0$, $w_2 = w_4 = 1$.

(d) Precision

$p = P(Y = +|\hat{Y} = +) = \frac{TP}{TP+FP}$, so $w_1 = w_2 = 1$, $w_3 = w_4 = 0$.

(e) Recall

Since Recall (r) is the same as sensitivity, it has the same weights.

(f) F_β

$$(\beta^2 + 1)rp = \frac{(\beta^2+1)TP^2}{(TP+FN)(TP+FP)}$$

$$r + \beta^2 p = \frac{TP}{(TP+FN)} + \beta^2 \frac{TP}{(TP+FP)} = \frac{TP(TP+FP+\beta^2(TP+FN))}{(TP+FN)(TP+FP)}$$

$$\begin{aligned} \frac{(\beta^2+1)rp}{r+\beta^2 p} &= \frac{(\beta^2+1)TP^2}{(TP+FN)(TP+FP)} \frac{(TP+FN)(TP+FP)}{TP(TP+FP+\beta^2(TP+FN))} \\ &= \frac{(\beta^2+1)TP}{(\beta^2+1)TP+FP+\beta^2 FN} \end{aligned}$$

So the weights are $w_1 = \beta^2 + 1,\ w_2 = 1,\ w_3 = \beta^2,\ w_4 = 0$.

3. Split `germancredit.csv` into 70% training and 30% test data.

(a) Fit a naive Bayes classifier for predicting default, and calculate accuracy, sensitivity, specificity, precision, and F_1 measure on test data.

Accuracy: 0.7533333

Sensitivity: 0.5777778

Specificity: 0.8285714

Precision: 0.5909091

F_1: 0.5842697

(b) Find the probability threshold p_0 that optimizes the F_1 measure on the training data.

The optimal value for p_0 is 0.28

(c) Recalculate the accuracy, sensitivity, specificity, precision, and F_1 measure on the test data using the new probability threshold.

Using $p_0 = 0.28$, the five measures calculated in part 3a were computed as shown below.

Accuracy: 0.4649682

Sensitivity: 0.8111111

Specificity: 0.0000000

Precision: 0.5214286

F_1: 0.6347826

```
#Data Mining hw 9
library(e1071)

# load and split the gernamcredit.csv into 70% and 30% training and test sets
gcred <- read.table("~/Dropbox/Tarleton/data_mining/dfiles/germancredit.csv",
                     header = T, sep=',')

#since we're trying to predict when there IS a default,
# set this to be positive value
gcred$Default <- as.factor(gcred$Default)

splitset <-splitdata(gcred, 0.7, FALSE)
train_i <- splitset$train

# Fit a naive Bayes classifier for predicting default, and calculate accuracy,
# sensitivity, specificity, precision, and F_1 measure on test data

modelD <- naiveBayes(Default~., gcred[train_i,])
predD <- predict(modelD, gcred[-train_i,])
pphat <- predict(modelD, gcred[-train_i,],type='raw')[,2]
Dtable <- matrix(rep(0, 4), ncol=2, nrow=2)
rownames(Dtable) <- c('1','0')
colnames(Dtable) <- c('1','0')
Dtable <- as.table(Dtable)
testpreddef <- (pphat >= 0.5) * 1

Dtable[1, 1] <- sum((testpreddef == 1) & (gcred$Default[-train_i] == '1')) #TP
```

```
Dtable[2, 1] <- sum((testpreddef == 1) & (gcred$Default[-train_i] != '1')) #FP
Dtable[1, 2] <- sum((testpreddef != 1) & (gcred$Default[-train_i] == '1')) #FN
Dtable[2, 2] <- sum((testpreddef != 1) & (gcred$Default[-train_i] != '1')) #TN

w1 <- c(accuracy(Dtable), sensitivity(Dtable),
        specificity(Dtable), precision(Dtable), F1(Dtable))

# find the probability threshold p0 that optimizes the F1 measure
# on the training data

predD1 <- predict(modelD, gcred[train_i,])
phat <- predict(modelD, gcred[train_i,], type='raw')[,2]

idx = seq(from=0.1, to = 0.9, by = 0.01)
F1acc = rep(-1, length(idx))
c = 1

mytable <- matrix(rep(0, 4),ncol=2,nrow=2)
colnames(mytable) <- c('1','0')
rownames(mytable) <- c('1','0')
mytable <- as.table(mytable)

for(p0 in idx){

  trainpreddef <- (phat >= p0) * 1
  mytable[1,1] <- sum( (trainpreddef == 1) & (gcred$Default[train_i] == '1') )
  mytable[2,1] <- sum( (trainpreddef == 1) & (gcred$Default[train_i] != '1') )
```

```
  mytable[1,2] <- sum( (trainpreddef != 1) & (gcred$Default[train_i] == '1') )

  F1acc[c] <- F1( mytable )
  c = c + 1
}
p0 <- idx[which.max(F1acc)]


# recalculate the accuracy, sensitivity, specificity, precision,
# and F1 measure on the test data using the new probability threshold.

nphat <- predict(modelD, gcred[-train_i,], type='raw')[,2]

testpreddef <- (nphat >= p0) * 1
mytable[1,1] <- sum( (testpreddef == 1) & (gcred$Default[-train_i] == '1') )
mytable[2,1] <- sum( (testpreddef == 1) & (gcred$Default[-train_i] != '1') )
mytable[1,2] <- sum( (testpreddef != 1) & (gcred$Default[-train_i] == '1') )

w2 <- c(accuracy(mytable), sensitivity(mytable),
        specificity(mytable), precision(mytable), F1(mytable))
```