Methods Homework 5

Tim Vigers

October 7, 2018

```r
# Load the libraries.
library(Epi)
library(knitr)
```

Evaluating diagnostic tests

a. Calculate sensitivity and specificity for CMMS Score <= 20

```r
# Make the data table.
cmms <- as.data.frame(matrix(c(0,2,0,1,3,4,9,5,16,3,18,1),ncol = 2,byrow = T))
cols<- c("non.demented","demented")
rows <- c("0-5","6-10","11-15","16-20","21-25","26-30")
colnames(cmms) <- cols
rownames(cmms) <- rows
cmms <- cmms[,c("demented","non.demented")]
kable(cmms)
```

	demented	non.demented
0-5	2	0
6-10	1	0
11-15	4	3
16-20	5	9
21-25	3	16
26-30	1	18

```r
# Convert to 2 x 2 table using <= 20 cutoff.
cmms20 <- cbind(c(sum(cmms$demented[1:4]),sum(cmms$demented[5:6])),
                c(sum(cmms$non.demented[1:4]),sum(cmms$non.demented[5:6])))
colnames(cmms20) <- cols
rownames(cmms20) <- c("<= 20",">20")
kable(cmms20)
```

	non.demented	demented
<= 20	12	12
>20	4	34

```r
# Sensitivity = a / (a + c) * 100, specificity = d / (d + b) * 100
sens20 <- (cmms20[1,1] / (sum(cmms20[,1])))*100
spec20 <- (cmms20[2,2] / (sum(cmms20[,2])))*100
sens20
```

```
## [1] 75
```

```
spec20
```

```
## [1] 73.91304
```

b. Calculate sensitivity and specificity for multiple CMMS Score cutoffs

```
# Make empty vectors for sensitivity and specificity.
spec <- NULL
sens <- NULL
# Use a for loop to calculate each cutoff.
for (r in 1:nrow(cmms)) {
  if (r<nrow(cmms)){
    new.table <- cbind(c(sum(cmms$demented[1:r],na.rm = TRUE),
                         sum(cmms$demented[(r+1):6],na.rm = TRUE)),
                       c(sum(cmms$non.demented[1:r],na.rm = TRUE),
                         sum(cmms$non.demented[(r+1):6],na.rm = TRUE)))
  } else {
    new.table <- cbind(c(sum(cmms$demented[1:r],na.rm = TRUE),0),
                       c(sum(cmms$non.demented[1:r],na.rm = TRUE),0))
  }
  sens <- c(sens,(new.table[1,1] / (sum(new.table[,1]))))
  spec <- c(spec,(new.table[2,2] / (sum(new.table[,2]))))
}
# Put results in a nice table.
sens.spec.table <- as.data.frame(matrix(nrow = 6,ncol = 3))
colnames(sens.spec.table) <- c("Cutoff","Sensitivity","Specificity")
sens.spec.table$Cutoff <- c("<=5", "<=10", "<=15", "<=20", "<=25","<=30")
sens.spec.table$Sensitivity <- sens
sens.spec.table$Specificity <- spec
kable(sens.spec.table)
```

Cutoff	Sensitivity	Specificity
<=5	0.1250	1.0000000
<=10	0.1875	1.0000000
<=15	0.4375	0.9347826
<=20	0.7500	0.7391304
<=25	0.9375	0.3913043
<=30	1.0000	0.0000000

c. False positives and negatives

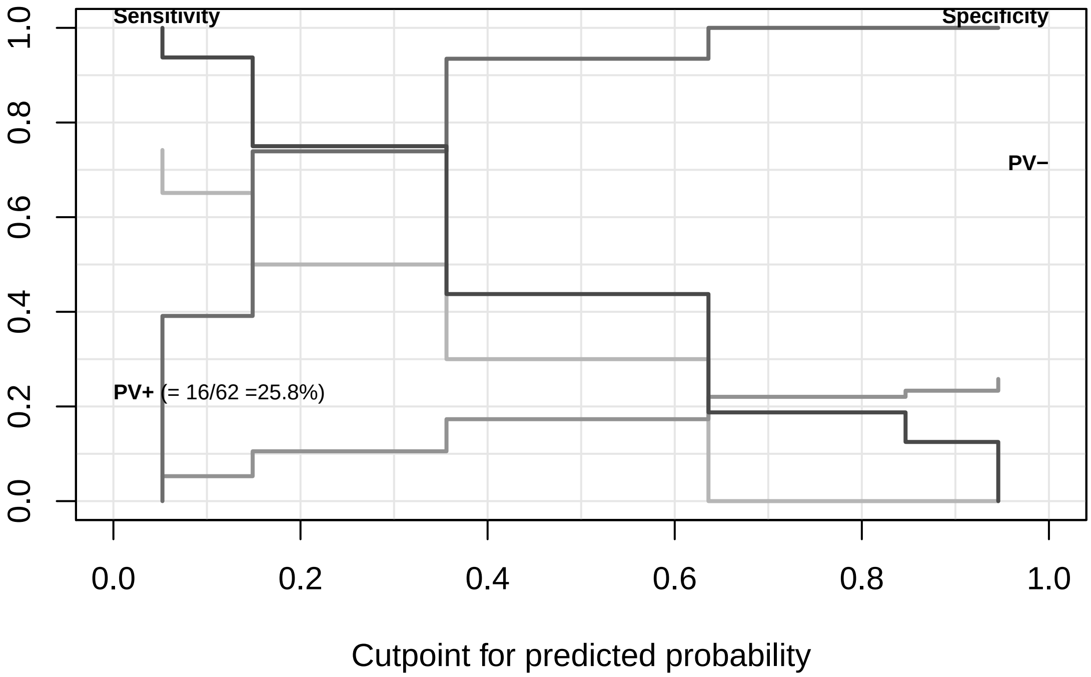
A false positive is when a test indicates the presence of a condition (in this case dementia), when the condition actually isn't there. False negatives are when the condition is there, but the test result is negative. The consequences of these depend a little bit on what dementia treatment involves, but in general a false negative is probably worse. If someone tested positive but did not actually have dementia, then they might start an unnecessary treatment. This could be bad, but is probably not as bad as someone testing negative and not getting any treatment when they do in fact have dementia (the false negative). So, I think we would want to optimize our cutoffs to reduce false negatives, which is the same as increasing sensitivity (1 - sensitivity = false negative rate). Then, for those that do test positive, you would want to do some additional testing to make sure they do indeed have dementia before starting treatment.

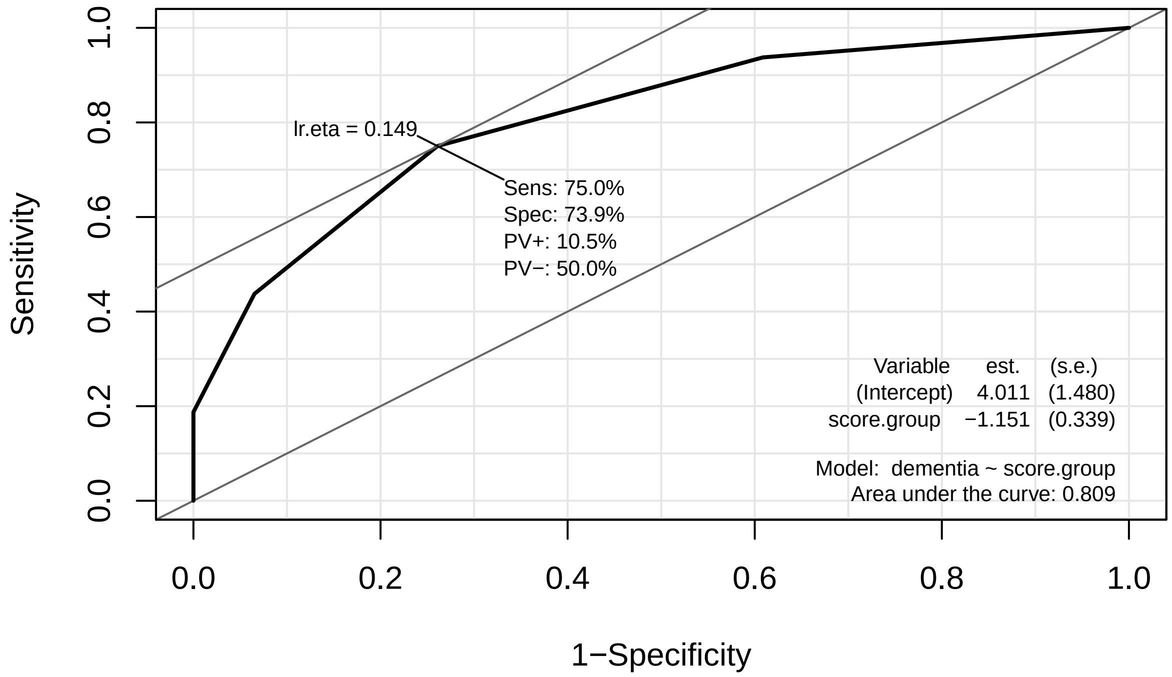
d. Choose a cutoff based on the sensitivity/specificity table.

If you assume that both false negatives and false positives are equally bad, then I think the <= 20 cutoff is a good one, as it maximizes both. If you wanted to maximize sensitivity, then you might say that anyone with a score <= 30 should get further testing. However, this would make the new test pretty useless, because everyone would need further testing.

e. ROC curve

```r
# Make a dataframe where each row is one person in the original dataset.
roc.data <- as.data.frame(matrix(nrow = sum(cmms),ncol = 2))
colnames(roc.data) <- c("score.group","dementia")
# Convert score groups into numbers 1-6, fill in score group column based on
# the total for the corresponding row in the original data.
roc.data$score.group <- c(rep(1,sum(cmms[1,])),
                          rep(2,sum(cmms[2,])),
                          rep(3,sum(cmms[3,])),
                          rep(4,sum(cmms[4,])),
                          rep(5,sum(cmms[5,])),
                          rep(6,sum(cmms[6,])))
# Dementia = 1, non-dementia = 0
roc.data$dementia <- c(rep(0,cmms$non.demented[1]),rep(1,cmms$demented[1]),
                       rep(0,cmms$non.demented[2]),rep(1,cmms$demented[2]),
                       rep(0,cmms$non.demented[3]),rep(1,cmms$demented[3]),
                       rep(0,cmms$non.demented[4]),rep(1,cmms$demented[4]),
                       rep(0,cmms$non.demented[5]),rep(1,cmms$demented[5]),
                       rep(0,cmms$non.demented[6]),rep(1,cmms$demented[6]))
# Run the Epi package's ROC function.
roc <- ROC(form = dementia ~ score.group,data = roc.data)
```

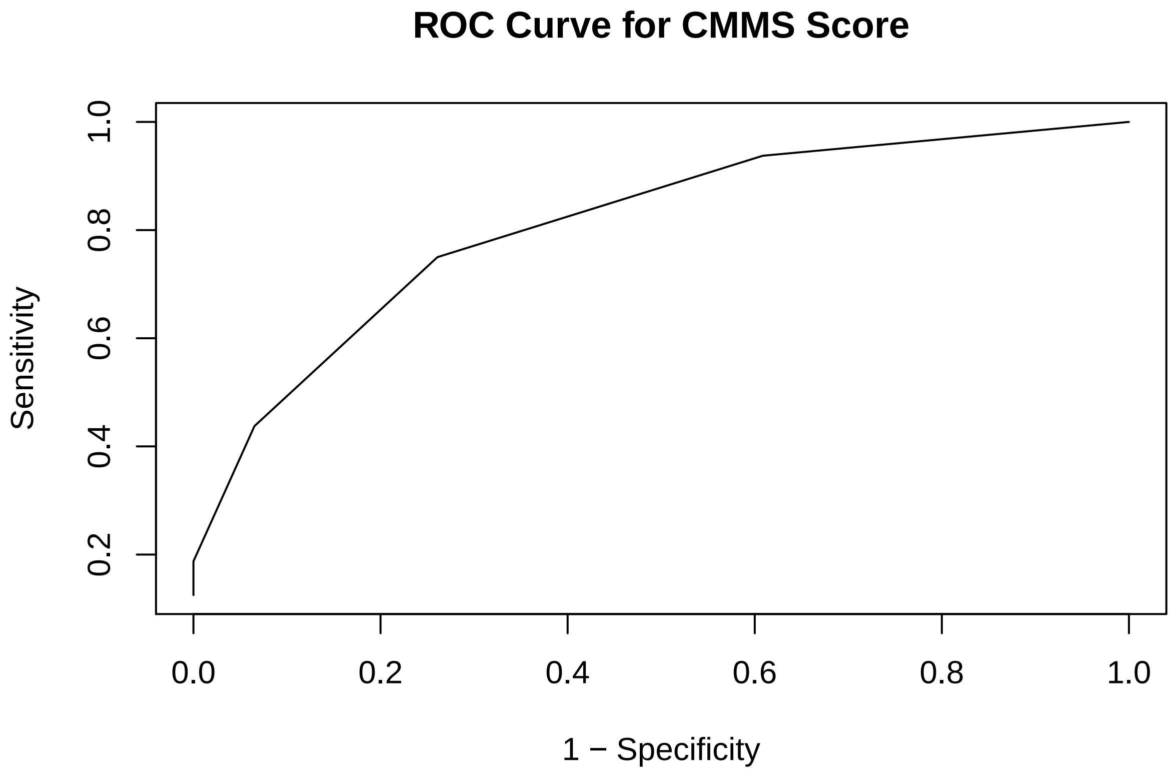

Sensitivity
Specificity
PV−
PV+ (= 16/62 =25.8%)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Cutpoint for predicted probability



The AUC of a ROC curve can range from 0 to 1, with AUC = 1 meaning the test is perfectly accurate, and AUC = 0.05 meaning the test is no better than random chance. So, this test appears to predict dementia with about 80% accuracy. In other words, if you have a pair of patients with and without dementia, there's an 80% chance that the person with dementia will have a worse CMMS score.

f. ROC plot using part B.

```
# Plot the table from part B, with lines connecting the points.
plot((1 - sens.spec.table$Specificity),sens.spec.table$Sensitivity,type = "l",xlab = "1 - Specificity",
```



```
# Calculate AUC with the trapezoidal rule.
AUC <- 0
for (v in 2:nrow(sens.spec.table)) {
  a <- sens.spec.table$Sensitivity[v-1]
  b <- sens.spec.table$Sensitivity[v]
  h <- (1 - sens.spec.table$Specificity[v]) -
    (1 - sens.spec.table$Specificity[v-1])
  trap <- ((a+b)/2)*h
  AUC <- AUC + trap
}
AUC
```

```
## [1] 0.8091033
```

This AUC is exactly the same as the one calculated by the Epi package.

g. PPV and NPV

$$PPV = \frac{\text{(sensitivity)(prior prob)}}{\text{(sensitivity)(prior prob) + (1-specificity)(1-prior prob)}}$$

$$NPV = \frac{\text{(specificity)(prior prob)}}{\text{(1 - sensitivity)(prior prob)+(specificity)(1 - prior prob)}}$$

```
# Sensitivity and specificity values at <= 20 cutoff.
spec <- sens.spec.table$Specificity[4]
sens <- sens.spec.table$Sensitivity[4]
```

```r
# For 10% prevalence
prior <- 0.1
PPV1 <- (sens * prior)/((sens*prior)+((1-spec) * (1-prior)))
PPV1
```

```
## [1] 0.2421053
```

```r
NPV1 <- (spec * (1-prior))/(((1-sens)*prior)+(spec*(1-prior)))
NPV1
```

```
## [1] 0.9637795
```

```r
# For 40% prevalence
prior <- 0.4
PPV2 <- (sens * prior)/((sens*prior)+((1-spec) * (1-prior)))
PPV2
```

```
## [1] 0.6571429
```

```r
NPV2 <- (spec * (1-prior))/(((1-sens)*prior)+(spec*(1-prior)))
NPV2
```

```
## [1] 0.816
```

For the 10% prevalence rate, a PPV of about 0.24 means that if a person tests positive based on CMMS (in this case a score <=20), there's about a 24% chance they actually have dementia. If the prevalence rate is 40%, then a positive test means the person has about a 66% chance of actually having dementia. PPV is highly dependent on prior probability, so as the prevalence increases, so does the predictive power of the test, which makes some intuitive sense.

For the 10% prevalence rate, an NPV of 0.96 means that if someone tests negative based on CMMS (score >20), there's about a 96% chance that they actually don't have the disease. At the higher prevalance rate this probability decreases to about 82%. Like with PPV, NPV is highly dependent on prior probability. But as prevalence goes up, NPV decreases.

h. Likelihood ratios

$$LR+ = \frac{\text{sensitivity}}{\text{1 - specificity}}$$

$$LR- = \frac{\text{1-sensitivity}}{\text{specificity}}$$

$$\text{Posterior odds of disease} = \text{prior odds of disease} * \text{LR+}$$

$$\text{Posterior odds of no disease} = \text{prior odds of no disease} * \frac{1}{\text{LR-}}$$

```r
# Calculate likelihood ratios.
sens.spec.table$lr.plus <-
  sens.spec.table$Sensitivity / (1 - sens.spec.table$Specificity)
sens.spec.table$lr.minus <-
  (1 - sens.spec.table$Sensitivity) / sens.spec.table$Specificity
# Calculate posterior odds for prevalence 0.3.
prior <- 0.3
sens.spec.table$post.odds <- prior * sens.spec.table$lr.plus
sens.spec.table$post.odds.no <- (1 - prior) * (1/sens.spec.table$lr.minus)
kable(sens.spec.table)
```

Cutoff	Sensitivity	Specificity	lr.plus	lr.minus	post.odds	post.odds.no
<=5	0.1250	1.0000000	Inf	0.8750000	Inf	0.8000000
<=10	0.1875	1.0000000	Inf	0.8125000	Inf	0.8615385
<=15	0.4375	0.9347826	6.708333	0.6017442	2.0125000	1.1632850
<=20	0.7500	0.7391304	2.875000	0.3382353	0.8625000	2.0695652
<=25	0.9375	0.3913043	1.540179	0.1597222	0.4620536	4.3826087
<=30	1.0000	0.0000000	1.000000	NaN	0.3000000	NaN

i.

Based on the criteria that LR+ >= 5 "rules in" disease and LR- <= 0.2 "rules out," none of these cutoffs are good for both.

ii.

Posterior odds of disease means that after a positive test result, an individual's odds are X:1 in favor of them having the disease. So for example at a <= 15 cutoff, a positive result means the patient has odds of about 2 to 1 that they have the disease. As the cutoff increases, the posterior odds decrease, which makes some intuitive sense.

Similarly, posterior odds of no disease means that after a negative test result, an individual's odds are X:1 in favor of them not having the disease. So again at a cutoff of <=15, a negative test result means the patient has odds of 1.16 to 1 of not having the disease. As the cutoff increases, the posterior odds of no disease also increase, which makes intuitive sense as well.