Lecture 19:
Classification (cont.)

Big Data and Machine Learning for Applied Economics
Econ 4676

Ignacio Sarmiento-Barbieri

Universidad de los Andes

October 20, 2020

Announcements

- **Problem Set 1 is graded, I'll be uploading the graded version to Github**
- To help with the grading and improve organization, Jacob created a demo repo and a rubric. Please follow it!
- **Problem Set 2 is due next Thursday September 22 at 11:00**
- At some point this afternoon or tomorrow morning I'll send presentation assignments
- You should consider class presentations as mini-seminars, just 2-5 minutes using one or two transparencies
- Attempt to make a concise interpretation of the relevant material, making effective use of supporting numerical and graphical evidence.

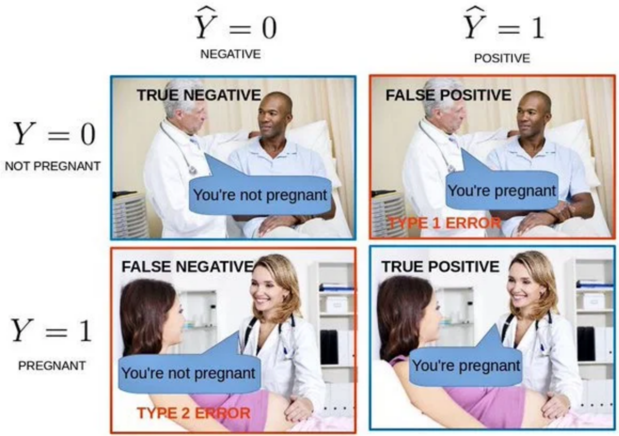
Agenda

Classification: Motivation

- Admit a student to *PEG* based on their grades and LoR
- Give a credit, based on credit history, demographics?
- Classifying emails: spam, personal, social based on email contents
- Aim is to classify y based on $X's$
- y can be
  - qualitative (e.g., spam, personal, social)
  - Not necessarily ordered
  - Not necessarily two categories, but will start with the binary case

Motivation

- Two states of nature $y \rightarrow i \in \{0, 1\}$
- Two actions $(\hat{y}) \rightarrow j \in \{0, 1\}$



Source: https://dzone.com/articles/understanding-the-confusion-matrix

Probability, Cost, and Classification

- Under a 0-1 penalty the problem boils down to finding $p = PR(Y = 1|X)$
- We then predict 1 if $p > 0.5$ and 0 otherwise (Bayes classifier)
- We can think 3 ways of finding this probability in binary cases
  - Knn
  - Logistic
  - LDA

Logit

We have a conditional probability

$$Pr(y = 1|X) = f(X'\beta) \tag{1}$$

Logistic regression uses a *logit* (sigmoid, softmax) link function

$$p(y = 1|X) = \frac{e^{X'\beta}}{1 + e^{X'\beta}} = \frac{exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}{1 + exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)} \tag{2}$$

Logit Demo

```r
set.seed(101010) #sets a seed
credit<-readRDS("credit_class.rds")
#70% train
indic<-sample(1:nrow(credit),floor(.7*nrow(credit)))
#Partition the sample
train<-credit[indic,]
test<-credit[-indic,]
head(credit)
```

```
##   Default duration amount installment age  history      purpose foreign  rent
## 1       0        6   1169           4  67 terrible goods/repair foreign FALSE
## 2       1       48   5951           2  22     poor goods/repair foreign FALSE
## 3       0       12   2096           2  49 terrible          edu foreign FALSE
## 4       0       42   7882           2  45     poor goods/repair foreign FALSE
## 5       1       24   4870           3  53     poor       newcar foreign FALSE
## 6       0       36   9055           2  35     poor          edu foreign FALSE
```

```r
dim(credit)
```

```
## [1] 1000    9
```

Logit Demo

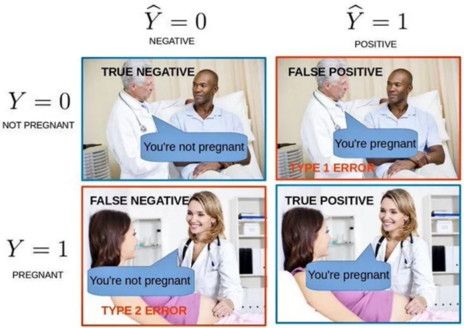
```r
mylogit <- glm(Default~duration + amount + installment + age
               + factor(history) + factor(purpose) + factor(foreign) + factor(rent),
               data = train, family = "binomial")
summary(mylogit)
```

```
##
## ...
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -3.285e-01  5.597e-01  -0.587 0.557264
## duration                    1.625e-02  9.538e-03   1.704 0.088369 .
## amount                      1.518e-04  4.325e-05   3.511 0.000447 ***
## installment                 3.335e-01  9.216e-02   3.619 0.000296 ***
## age                        -1.762e-02  8.851e-03  -1.990 0.046554 *
## factor(history)poor        -1.212e+00  3.126e-01  -3.876 0.000106 ***
## factor(history)terrible    -1.989e+00  3.552e-01  -5.598 2.17e-08 ***
## factor(purpose)usedcar     -1.813e+00  4.067e-01  -4.459 8.23e-06 ***
## factor(purpose)goods/repair -7.163e-01  2.254e-01  -3.177 0.001486 **
## factor(purpose)edu          1.207e-01  3.858e-01   0.313 0.754450
## factor(purpose)biz         -9.862e-01  3.440e-01  -2.867 0.004147 **
## factor(foreign)german      -2.057e+00  8.213e-01  -2.505 0.012254 *
## factor(rent)TRUE            7.554e-01  2.355e-01   3.208 0.001337 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ...
```

Logit Demo

```r
test$phat<- predict(mylogit, test, type="response")
test$Default_hat<-ifelse(test$phat>.5,1,0)
with(test,prop.table(table(Default,Default_hat)))
```

```
##          Default_hat
## Default           0          1
##       0 0.63666667 0.06666667
##       1 0.22666667 0.07000000
```



Linear Discriminant Analysis

Linear Discriminant Analysis

Reverend Bayes to the rescue: Bayes Theorem

$$p(Y=1|X)=\frac{f(X|Y=1)p(Y=1)}{m(X)} \tag{3}$$

with $m(X)$ is the marginal distribution of X, i.e.

$$m(X)=\int f(X|Y=1)p(Y=1)dy \tag{4}$$

Recall that there are two states of nature $y \rightarrow i \in \{0,1\}$

$$\begin{aligned} m(X) &= f(X|Y=1)p(Y=1)+f(X|Y=0)p(Y=0) \\ &= f(X|Y=1)p(Y=1)+f(X|Y=0)(1-p(Y=1)) \end{aligned} \tag{5}$$

Linear Discriminant Analysis

- This is basically an empirical Bayes approach
- We need to estimate $f(X|Y=1), f(X|Y=0)$ and $p(Y=1)$
  - Let's start by estimating $p(Y=1)$. We've done this before

$$p(Y=1) = \frac{\sum_{i=1}^{n} 1[Y_i = 1]}{N} \tag{6}$$

  - Next $f(X|Y=j)$ with $j=0,1$.
    - if we assume one predictor and $X|Y \sim N(\mu_j, \sigma_j)$
    - the problem boils down to estimating μ_j, σ_j
    - LDA makes it simpler, assumes $\sigma_j = \sigma \; \forall j$
    - then partition the sample in two $Y=0$ and $Y=1$, estimate the moments and get $\hat{f}(X|Y=j)$
  - Plug everything into the Bayes Rule and you're done

Linear Discriminant Analysis: Demo

$$p(Y=1) = \frac{\sum_{i=1}^{n} 1[Y_i = 1]}{N} \tag{7}$$

```
p1<-sum(train$Default)/dim(train)[1]
p1
```

```
## [1] 0.3014286
```

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i \tag{8}$$

```
mu1<-mean(train$duration[train$Default==1])
mu1
```

```
## [1] 24.78673
```

```
mu0<-mean(train$duration[train$Default==0])
mu0
```

```
## [1] 19.79346
```

Linear Discriminant Analysis: Demo

$$\hat{\sigma}^2 = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2 \tag{9}$$

```
g1<-sum((train$duration[train$Default==1]-mu1)^2)
g0<-sum((train$duration[train$Default==0]-mu0)^2)


sigma<-sqrt((g1+g0)/(dim(train)[1]-2))
```

$$\hat{f}_k \sim N(\hat{\mu}_k, \hat{\sigma}) \tag{10}$$

```
f1<-dnorm(test$duration,mean=mu1,sd=sigma)
f0<-dnorm(test$duration,mean=mu0,sd=sigma)
```

Linear Discriminant Analysis: Demo

```r
library("MASS")      # LDA
lda_simple <- lda(Default~duration, data = train)
lda_simple_pred<-predict(lda_simple,test)
names(lda_simple_pred)
```

```
## [1] "class"     "posterior" "x"
```

```r
posteriors<-data.frame(lda_simple_pred$posterior)
posteriors$hand<-f1*p1/(f1*p1+f0*(1-p1))
head(posteriors)
```

```
##           X0        X1      hand
## 1  0.8013656 0.1986344 0.1986344
## 3  0.7668614 0.2331386 0.2331386
## 14 0.6861792 0.3138208 0.3138208
## 16 0.6861792 0.3138208 0.3138208
## 28 0.7668614 0.2331386 0.2331386
## 33 0.7283950 0.2716050 0.2716050
```

Linear Discriminant Analysis
Extensions

- If we have k predictors?
- then $X|Y\ NM(\mu, \Sigma)$

$$f(X|Y=j) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu_j)'\Sigma_j(x-\mu_j) \tag{11}$$

- μ_j is the vector of the sample means in each partition $j = 0, 1$
- Σ_j is the matrix of variance and covariances of each partition $j = 0, 1$
- Can we lift normality?

Linear Discriminant Analysis

- Why is it call linear?
- Note

$$p > \frac{1}{2} \iff ln(\frac{p}{(1-p)}) \tag{12}$$

- Logit with one predictor

$$\beta_1 + \beta_2 X \tag{13}$$

- Classification: in the probability of space
- Discrimination: in the space of X
- $\beta_1 + \beta_2 X$ is the discrimination function for logit (it is lineal)

Linear Discriminant Analysis

- LDA?
- One predictor with $\sigma_0 = \sigma_1$ (equal variance)

$$p(Y=1|X) = \frac{f(X|Y=1)p(Y=1)}{f(X|Y=1)p(Y=1) + f(X|Y=0)(1-p(Y=1))} \tag{14}$$

- Then under the equal variance assumption

$$\frac{p(Y=1|X)}{1-p(Y=1|X} = \frac{f(X|Y=1)p(Y=1)}{f(X|Y=0)(1-p(Y=1))} \tag{15}$$

$$= \frac{p(Y=1)exp((x-\mu_1)^2)}{(1-p(Y=1))exp((x-\mu_0)^2)} \tag{16}$$

Linear Discriminant Analysis

- Taking logs

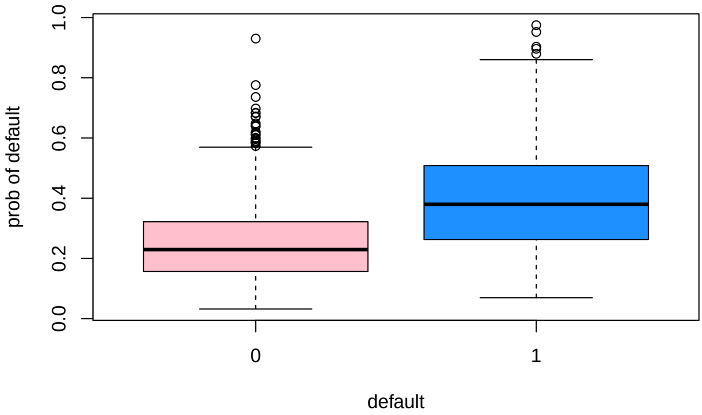
$$log\left(\frac{p(Y=1|X)}{1-p(Y=1|X}\right) = log(\frac{p(Y=1)}{(1-p(Y=1))} + (x-\mu_1)^2 - (x-\mu_0)^2 \tag{17}$$
$$= log(\frac{p(Y=1)}{(1-p(Y=1))} + \mu_1^2 - \mu_0^2 - 2(\mu_1-\mu_0)x \tag{18}$$
$$= \gamma_1 + \gamma_2 X \tag{19}$$

- under the assumption of equal variance the discrimination function is lineal
- Note: logit estimates γ_1 and γ_2

Misclassification Rates

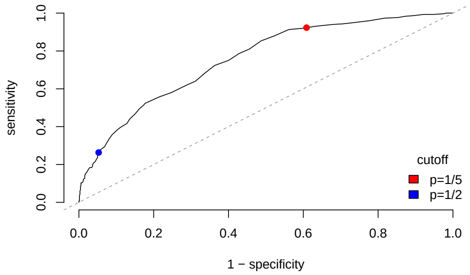
Misclassification Rates

- Predicted probabilities from Logit model



Misclassification Rates

- A classification rule, or cutoff, is the probability p at which you predict
  - $\hat{y}_i = 0$ if $p_i < p$
  - $\hat{y}_i = 1$ if $p_i < p$
- Measures of performance
  - *1-Specificity:* False Positive Rate, Type I error
  - *Sensitivity:* True Positive Rate, power, (1-Type II error)

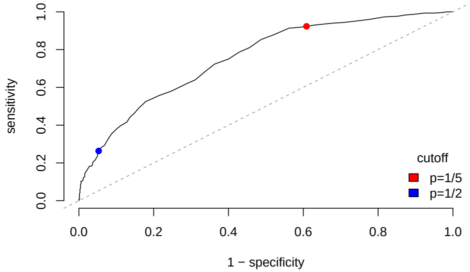


ROC

- ROC curve: Receiver operating characteristic curve
- ROC curve illustrates the trade-off of the classification rule
- Gives us the ability
  - Measure the predictive capacity of our model
  - Compare between models
- Some definitions
  - $P = \sum Y_i$ positives
  - $N = \sum(1 - Y_i)$ negatives
  - $T = P + N$ all observations
  - True Positives: $TP = \sum \hat{Y}_i Y_i$, True Positive Rate = $\frac{TP}{P}$
  - False Positives: $FP = \sum \hat{Y}_i (1 - Y_i)$, False Positive Rate = $\frac{FP}{N}$

ROC

- Binary Classifier: $\hat{Y}_i = 1[p_i > c], c \in [0, 1]$
- Bayes fixes $c = 0.5$
- Ideally $TPR = 1$ and $FPR = 0$
- ROC curve give us the locus of all possible TPR and FPR for all possible $c \in [0, 1]$



ROC

- ROC Properties
  - Has positive slope
    - In $(0,0)$, $c = 1$. When $c \downarrow$, $TP \uparrow$ and $FP \uparrow$. Then

$$TPR = \sum \frac{\hat{Y}_i Y_i}{P} \quad FPR = \sum \frac{\hat{Y}_i (1 - Y_i)}{T - P} \tag{20}$$

    - Is easy to show

$$TPR = \frac{\sum \hat{Y}_i}{P} - \frac{T - P}{P} FPR \tag{21}$$

    - ROC is the locus of all possible TPR and FPR for all possible $c \in [0, 1]$

$$TPR = \frac{\sum \hat{Y}_i(c)}{P} - \frac{T - P}{P} FPR(c) \tag{22}$$

Inverse Classifier

- ROC Properties
  - ROC curve is above the 45° line (TPR=FPR)
  - Note that

$$\hat{Y}_i^F = 1 - \hat{Y}_i \tag{23}$$

  - Recall that

$$TPR = \sum \frac{\hat{Y}_i Y_i}{P} \quad FPR = \sum \frac{\hat{Y}_i (1 - Y_i)}{T - P} \tag{24}$$

  - the inverse clasifivier would be

$$TPR^F = \sum \frac{(1 - \hat{Y}_i) Y_i}{P} \quad FPR^F = \sum \frac{(1 - \hat{Y}_i)(1 - Y_i)}{T - P} \tag{25}$$

  - Then $TPR - FPR = TPR^F - FPR^F$
  - If ROC is bellow the 45° line (TPR=FPR) then $FPR > TPR$. Given the above equality, the inverse classifier is above

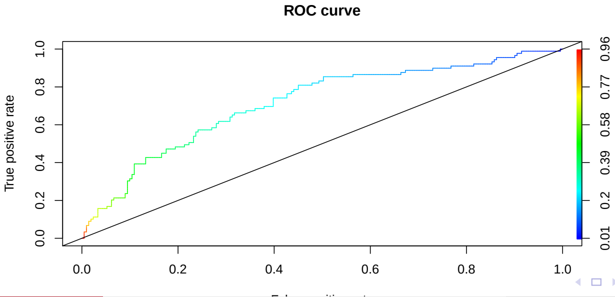
ROC: Summary

- Ideal ROC curve
- AUC: area under the curve, is like an R^2
- Help us compare between classifiers
- Dominated classifiers?
- Which c? Choose a max FPR

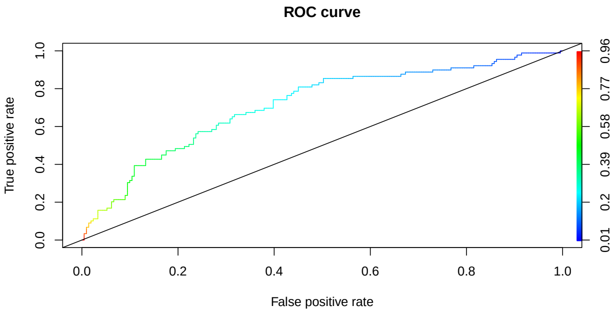
Roc Demo

```
library("ROCR") #Roc

mylogit <- glm(Default~duration + amount + installment + age
               + factor(history) + factor(purpose) + factor(foreign) + factor(rent),
               data = train, family = "binomial")
test$phat<- predict(mylogit, test, type="response")
pred <- prediction(test$phat, test$Default)
roc_ROCR <- performance(pred,"tpr","fpr")
plot(roc_ROCR, main = "ROC curve", colorize = T)
abline(a = 0, b = 1)
```



Roc Demo




```r
auc_ROCR <- performance(pred, measure = "auc")
auc_ROCR@y.values[[1]]
```

```
## [1] 0.714415
```

Roc Demo

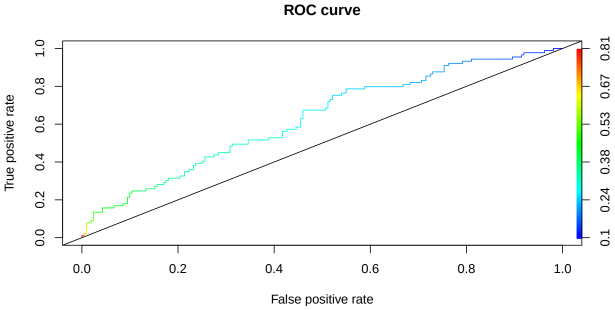
```r
mylda <- lda(Default~duration + amount + installment + age , data = train)
mylda
```

```
## Call:
## lda(Default ~ duration + amount + installment + age, data = train)
##
## Prior probabilities of groups:
##         0         1
## 0.6985714 0.3014286
##
## Group means:
##   duration   amount installment      age
## 0 19.79346 3062.888    2.885481 36.40900
## 1 24.78673 4057.791    3.109005 33.85782
##
## Coefficients of linear discriminants:
##                       LD1
## duration     0.0296041361
## amount       0.0002055164
## installment  0.4821242957
## age         -0.0386710882
```

Roc Demo

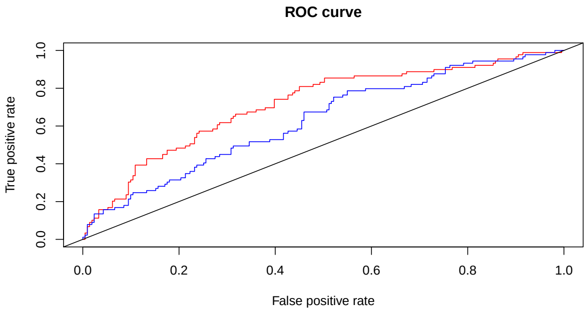
```
phat_mylda<- predict(mylda, test, type="response")
pred_mylda <- prediction(phat_mylda$posterior[,2], test$Default)

roc_mylda <- performance(pred_mylda,"tpr","fpr")
plot(roc_mylda, main = "ROC curve", colorize = T)
abline(a = 0, b = 1)
```



Roc Demo

```
plot(roc_ROCR, main = "ROC curve", colorize = FALSE, col="red")
plot(roc_mylda,add=TRUE, colorize = FALSE, col="blue")
abline(a = 0, b = 1)
```



Roc Demo

- Area under the curve (AUC)

```
auc_ROCR <- performance(pred, measure = "auc")
auc_ROCR_lda_simple <- performance(pred_mylda, measure = "auc")
auc_ROCR@y.values[[1]]
```

```
## [1] 0.714415
```

```
auc_ROCR_lda_simple@y.values[[1]]
```

```
## [1] 0.6291602
```

Review & Next Steps

- Review Classification:
  - KNN
    - Intuitive
    - Not very useful in practice, curse of dimensionality
  - Logit
  - Linear Discriminant Analysis
  - Misclassification Rates: ROC curve
  - QDA?
  - Multiple Classes?
- Next class: Problem Sets, Text Data!
- Questions? Questions about software?

Further Readings

- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Taddy, M. (2019). Business data science: Combining machine learning and economics to optimize, automate, and accelerate business decisions. McGraw Hill Professional.