Generative Models for Classification and Missclassification

Big Data y Machine Learning para Economía Aplicada

Ignacio Sarmiento-Barbieri

Universidad de los Andes

Agenda

Agenda

1. Recap
2. Generative Models for Classification
   - Discriminant Analysis
   - Naive Bayes
3. Misclassification Rates
   - ROC curve
4. Multiple Classes
   - KNN
   - Multinomial Logit

Recap

- We observe (y_i, X_i) $i = 1, \ldots, n$
- Estimate Probabilities
  - Logit

$$p_i = \frac{e^{X_i\beta}}{1 + e^{X_i\beta}} \tag{1}$$

  - get β
- Prediction
  - Logit, with the $\hat{\beta}$

$$\hat{p}_i = \frac{e^{X_i\hat{\beta}}}{1 + e^{X_i\hat{\beta}}} \tag{2}$$

- Classification

$$\hat{Y}_i = 1[\hat{p}_i > c] \tag{3}$$

Agenda

1. Recap
2. Generative Models for Classification
   - Discriminant Analysis
   - Naive Bayes
3. Misclassification Rates
   - ROC curve
4. Multiple Classes
   - KNN
   - Multinomial Logit

Linear Discriminant Analysis

Reverend Bayes to the rescue: Bayes Theorem

$$Pr(Y = 1|X) \tag{4}$$

Linear Discriminant Analysis, k=1

Linear Discriminant Analysis, k¿1

Extensions

- If we have $k > 1$ predictors?
- then $X|Y \sim NM(\mu, \Sigma)$

$$f(X|Y=j) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu_j)'\Sigma_j(x-\mu_j) \tag{5}$$

- 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 $j = 0, 1$

Linear Discriminant Analysis

- Why is it called linear?
- Note

$$p > \frac{1}{2} \iff ln(\frac{p}{(1-p)}) \tag{6}$$

- Logit with one predictor

$$\beta_1 + \beta_2 X \tag{7}$$

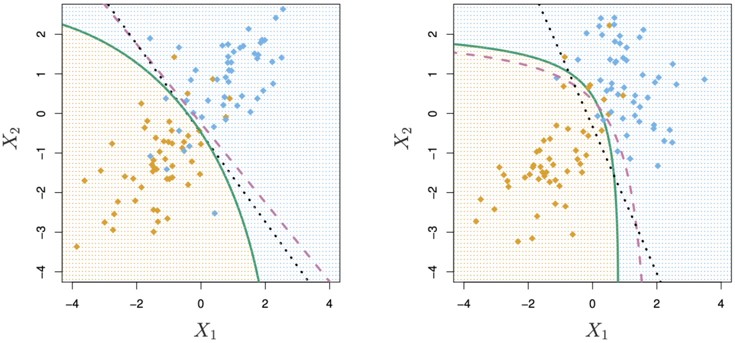
- Classification: in the probability of space
- Discrimination: in the space of X
- $\beta_1 + \beta_2 X$ is the discrimination function for logit (it is lineal)

Linear Discriminant Analysis

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

Quadratic Discriminant Analysis

- QDA assumes diferent variances for the components



Naive Bayes

$$Pr(Y=1|X) = \frac{f(X|Y=1)\pi(Y=1)}{f(X|Y=1)\pi(Y=1) + f(X|Y=0)(1-\pi(Y=1))} \quad (8)$$

- $\pi(Y=1)$
- $f(X|Y=1)$

Naive Bayes

- NB assumes independence

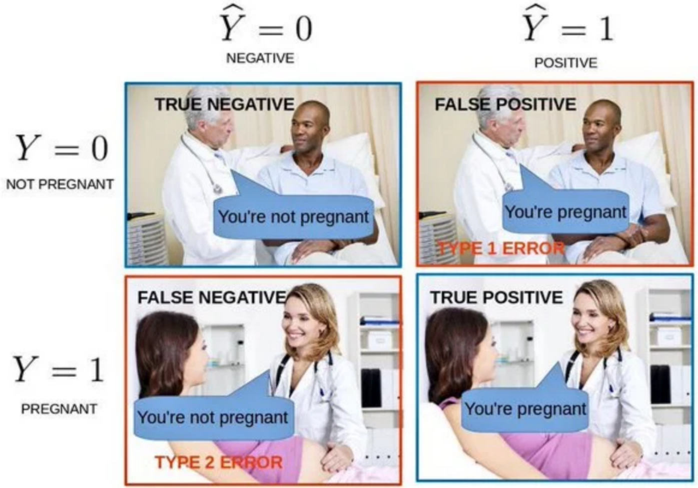
$$f(X|Y=1) = f(x_1|Y=1) \times \cdots \times f(x_k|Y=1) \tag{9}$$

Example: Default

photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/

Agenda

Misclassification Rates



Misclassification Rates

$$
\begin{array}{cccc}
 & & \hat{y}_i & \\
 & & 0 & 1 \\
\multirow{2}{*}{y_i} & 0 & TN & FP \\
 & 1 & FN & TP
\end{array}
$$

- We have several types of error associated with this that we can use as a measure of performance

ROC

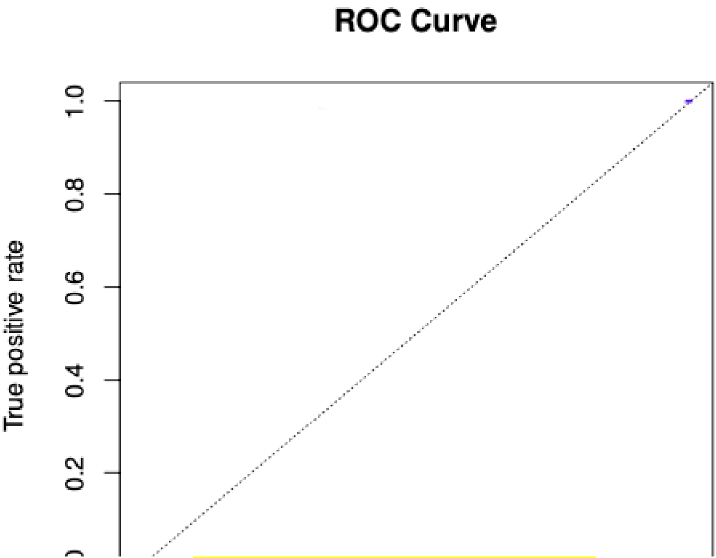
		$\hat{y}_i$	
		0	1
y_i	0	TN	FP
	1	FN	TP

- A classification rule, or cutoff, is the probability p at which you predict
  - $\hat{y}_i = 0$ if $p_i < c$
  - $\hat{y}_i = 1$ if $p_i > c$
- Bayes classifier $c = 0.5$
- Changing c changes predictions, changes FP and FN
- There is a trade-off: reducing one error increases the other

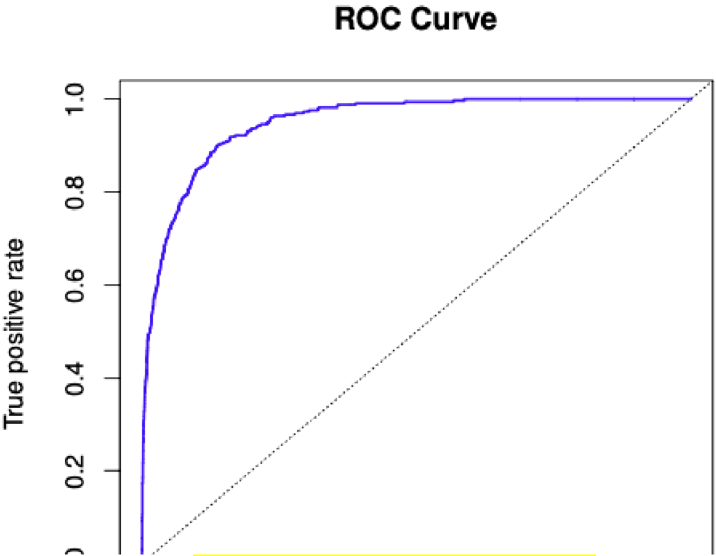
ROC

- 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

ROC



ROC



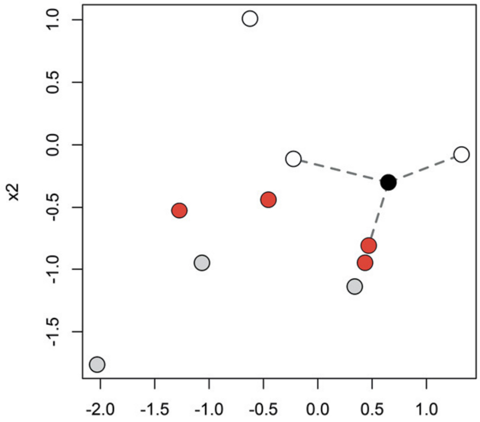
Example: Default

photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/

Agenda

K-Nearest Neighbors

- What happens when we have to predict multiple outcomes?
- K nearest neighbor (K-NN) algorithm predicts class $\hat{y}$ for x by asking *What is the most common class for observations around x?*



K-Nearest Neighbors

- K nearest neighbor (K-NN) algorithm predicts class $\hat{y}$ for x by asking
  What is the most common class for observations around x?
- Algorithm: given an input vector x_f where you would like to predict the class label
  - Find the K nearest neighbors in the dataset of labeled observations, $\{x_i, y_i\}_{i=1}^n$, the most common distance is the Euclidean distance:

$$d(x_i, x_f) = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{fj})^2} \tag{10}$$

  - This yields a set of the K nearest observations with labels:

$$[x_{i1}, y_{i1}], \ldots, [x_{iK}, y_{iK}] \tag{11}$$

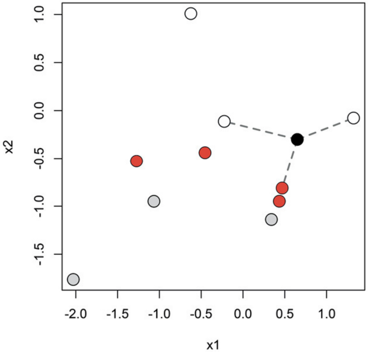
  - The predicted class of x_f is the most common class in this set

$$\hat{y}_f = mode\{y_{i1}, \ldots, y_{iK}\} \tag{12}$$

K-Nearest Neighbors

- There are some major problems with practical implications
  - Knn predictions are unstable as a function of K

$$
\begin{aligned}
K = 1 &\implies \hat{p}(white) = 0 \\
K = 2 &\implies \hat{p}(white) = 1/2 \\
K = 3 &\implies \hat{p}(white) = 2/3 \\
K = 4 &\implies \hat{p}(white) = 1/2
\end{aligned}
$$



Source: Taddy (2019)

K-Nearest Neighbors

- There are some major problems with practical implications
  - Knn predictions are unstable as a function of K
  - This instability of prediction makes it hard to choose the optimal K and cross validation doesn't work well for KNN
  - Since prediction for each new x requires a computationally intensive counting, KNN is too expensive to be useful in most big data settings.
  - KNN is a good idea, but too crude to be useful in practice

The multinomial logit model: Intuition

- The MNLM can be thought of as simultaneously fitting binary logits for all comparisons among the alternatives.
- For example,
  - We have a categorical variable with the outcomes for Democrat, for Independent, and for Republican.
  - Assume that there is one independent variable measuring income in $1,000s$.
  - We can examine the effect of income on party by fitting three binary logits,

The multinomial logit model: Intuition

- The MNLM can be thought of as simultaneously fitting binary logits for all comparisons among the alternatives.

$$
\begin{aligned}
\ln \frac{Pr(D|X)}{Pr(I/X)} &= \beta_{0,D|I} + \beta_{1,D|I} Income \\
\ln \frac{Pr(R|X)}{Pr(I/X)} &= \beta_{0,R|I} + \beta_{1,R|I} Income \\
\ln \frac{Pr(D|X)}{Pr(R/X)} &= \beta_{0,D|R} + \beta_{1,D|R} Income
\end{aligned} \tag{13}
$$

- where the subscripts to the $\beta's$ indicate which comparison is being made.

The multinomial logit model: Intuition

- These logits include redundant info

$$\ln \frac{Pr(D|X)}{Pr(I/X)} - \ln \frac{Pr(R|X)}{Pr(I/X)} = \ln \frac{Pr(D|X)}{Pr(R/X)} \tag{14}$$

- which implies

$$\beta_{0,D|I} - \beta_{0,R|I} = \beta_{0,D|R} \tag{15}$$

$$\beta_{1,D|I} - \beta_{1,R|I} = \beta_{1,D|R} \tag{16}$$

- In general, with J alternatives, only $J-1$ binary logits need to be fit (minimal set)

The multinomial logit model: Intuition

	Binary		
	(1)	(2)	(3)
VARIABLES	dem_ind	rep_ind	dem_rep
income	-0.00249	0.0157***	-0.0184***
	(0.00355)	(0.00374)	(0.00230)
Constant	1.605***	0.659***	0.953***
	(0.149)	(0.162)	(0.105)
Observations	844	689	1,231

- Fitting the MNLM by fitting a series of binary logits is not optimal

The multinomial logit model: Intuition

	Binary		
	(1)	(2)	(3)
VARIABLES	dem_ind	rep_ind	dem_rep
income	-0.00249	0.0157***	-0.0184***
	(0.00355)	(0.00374)	(0.00230)
Constant	1.605***	0.659***	0.953***
	(0.149)	(0.162)	(0.105)
Observations	844	689	1,231

- Fitting the MNLM by fitting a series of binary logits is not optimal
  - Binary logit is based on a different sample.
  - It ignores the restricctions that are implicit in the definition of the MNLM

The multinomial logit model: formal statement

- Formally

$$\ln \Omega_{m|b}(X) = \ln \frac{Pr(y=m|X)}{Pr(y=b|X)} = X\beta_{m|b} \; for\, m = 1, \ldots, J \tag{17}$$

- were b is the base outcome (reference category)
- These J equations can be solved to compute the probabilities for each outcome

$$Pr(y=m|X) = \frac{exp(X\beta_{m|b})}{\sum_{j=1}^{J} exp(X\beta_{j|b})} \tag{18}$$

The multinomial logit model: Intuition

	Binary			Multinomial Logit		
	(1)	(2)	(3)			
VARIABLES	dem_ind	rep_ind	dem_rep	Democrat	Independent	Republican
income	-0.00249	0.0157***	-0.0184***	-0.00272		0.0152***
	(0.00355)	(0.00374)	(0.00230)	(0.00372)		(0.00366)
Constant	1.605***	0.659***	0.953***	1.613***		0.678***
	(0.149)	(0.162)	(0.105)	(0.153)		(0.160)
Observations	844	689	1,231	1,382	1,382	1,382