Ch 4.3 - Logistic Regression

Lecture 10 - CMSE 381

Prof. Elizabeth Munch

Michigan State University
::
Dept of Computational Mathematics, Science & Engineering

Fri, Sep 20, 2024

Announcements

Lec #	Date			Reading	HW
1	Mon	8/26	Intro / First day stuff / Python Review Pt 1	1	
2	Wed	8/28	What is statistical learning?	2.1	
3	Wed	9/4	Assessing Model Accuracy	2.2.1, 2.2.2	
4	Fri	9/6	Linear Regression	3.1	HW #1 Due Sun 9/8
5	Mon	9/9	More Linear Regression	3.1	
6	Wed	9/11	Multi-linear regression	3.2	
7	Fri	9/13	Probably more linear regression	3.3	Hw #2 Due Dun 9/15
8	Mon	9/16	Last of the linear regression		
9	Wed	9/18	Intro to classification, Bayes classifier, KNN classifier	2.2.3	
10	Fri	9/20	Logistic Regression	4.1, 4.2, 4.3.1-3	Hw #3 Due Sun 9/22
11	Mon	9/23	Multiple Logistic Regression / Multinomial Logistic Regression	4.3.4-5	
	Wed	9/25	***Project Day & Review***		
	Fri	9/27	**Midterm #1**		

Announcements:

- Homework #3 Due Sunday on Crowdmark
- Wednesday - Review day
  - Nothing prepped
  - Bring your questions
- Friday - Exam #1
  - Bring 8.5x11 sheet of paper
  - Handwritten both sides
  - Anything you want on it, but must be your work
  - You will turn it in

Covered in this lecture

Last Time:

- Classification basics
- Bayes classifier
- KNN classifier

This time:

- Logistic Regression

Section 1

Review from last time

Error rate

- Training data:
  $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ with y_i qualitative
- Estimate $\hat{y} = \hat{f}(x)$
- Indicator variable

Training error rate:

$$\frac{1}{n}\sum_{i=1}^{n} \mathrm{I}(y_i \neq \hat{y}_i)$$

Test error rate:

$$\mathrm{Ave}(\mathrm{I}(y_0 \neq \hat{y}_0))$$

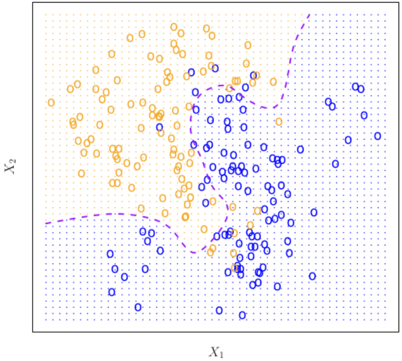
Best ever classifier

We can't have nice things

Bayes Classifier:
Give every observation the highest probability class given its predictor variables

$$\Pr(Y = j \mid X = x_0)$$

Bayes Decision Boundary



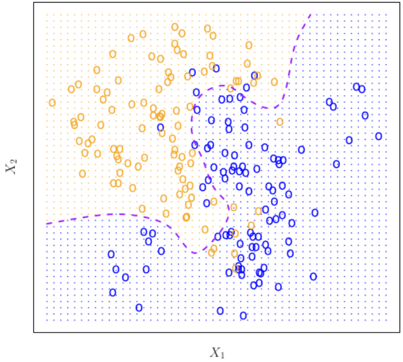
Bayes error rate

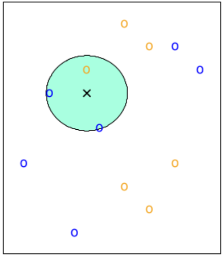
- Error at $X = x_0$

$$1 - \max_j \Pr(Y = j \mid X = x_0)$$

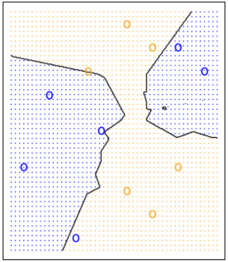
- Overall Bayes error:

$$1 - E\left(\max_j \Pr(Y = j \mid X = x_0)\right)$$

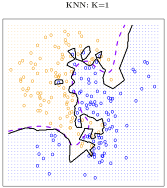


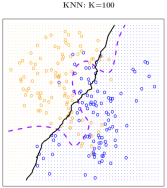


$K = 3$



decision boundary

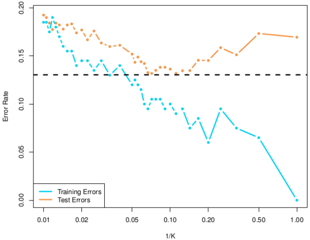




- Fix K positive integer
- $N(x)$ = the set of K closest neighbors to x
- Estimate conditional proabability

$$\Pr(Y = j \mid X = x_0) = \frac{1}{K} \sum_{i \in N(x_0)} \mathrm{I}(y_i = j)$$

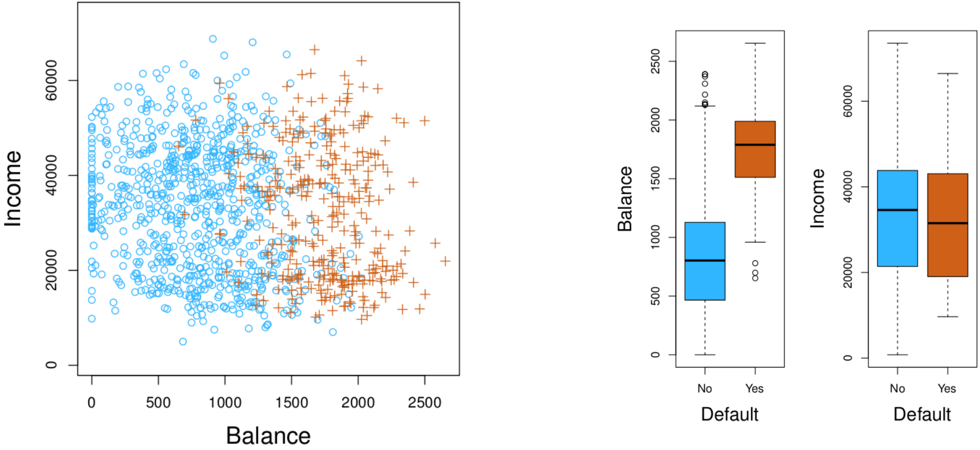
- Pick j with highest value



Section 2

Logistic Regression

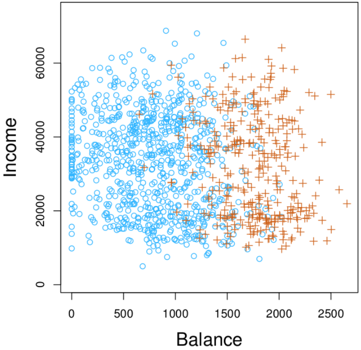
Simulated `Default` data set



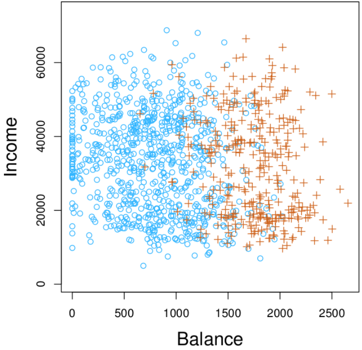
What is classification

- Classification: When the response variable is qualitative
- Goal: Model the probability that Y belongs to a particular category

$$p(\texttt{balance}) = \Pr(\texttt{default} = \texttt{yes} \mid \texttt{balance})$$



Goal for `Balance` data set



Goal: Model the probability that Y belongs to a particular category

Ex.

$\Pr(\texttt{default} = \texttt{yes} \mid \texttt{balance})$

Let's just use regression!

JK that's a bad idea

Bad idea:

- Set Y to be a dummy variable taking values in $\{0, 1, 2, \cdots\}$
- Run regression, and choose k based on what integer value $\hat{y}$ is closest to

Ex.

$$Y = \begin{cases} 1 & \text{if stroke} \\ 2 & \text{if drug overdose} \\ 3 & \text{if epileptic seizure} \end{cases}$$

vs.

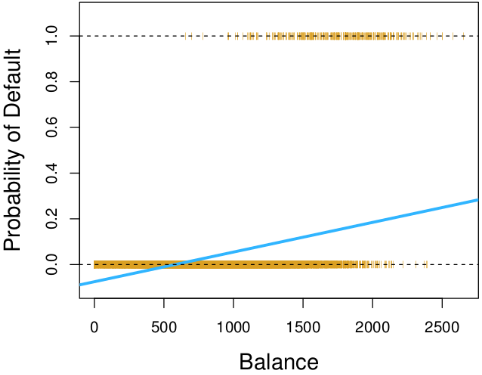
$$Y = \begin{cases} 1 & \text{if mild} \\ 2 & \text{if moderate} \\ 3 & \text{if severe} \end{cases}$$

Bad idea is still not a great idea for two levels

$$p(\texttt{balance}) = \Pr(\texttt{default} = \texttt{yes} \mid \texttt{balance})$$

$$Y = \begin{cases} 0 & \text{if not default} \\ 1 & \text{if default} \end{cases}$$

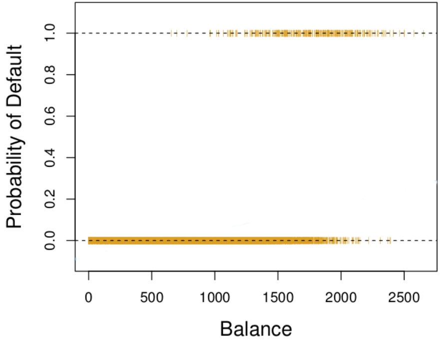
- Fit linear regression
- Predict default if $\hat{y} > 0.5$; not default otherwise



$$p(\texttt{balance}) = \beta_0 + \beta_1 \texttt{balance}$$

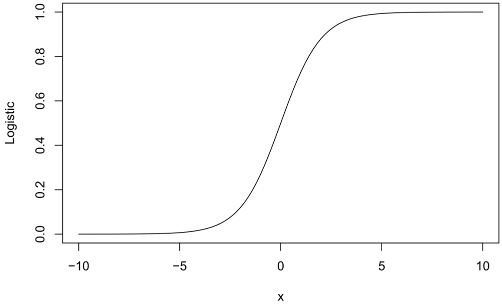
Approximating the probability

$$\Pr(\texttt{default} = \texttt{yes} \mid \texttt{balance})$$



Logistic function

$$y = \frac{e^x}{1 + e^x}$$

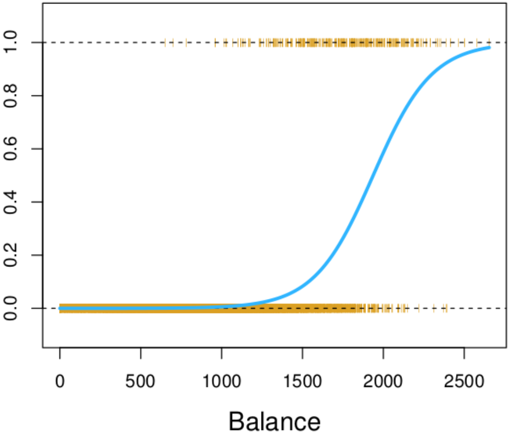


$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

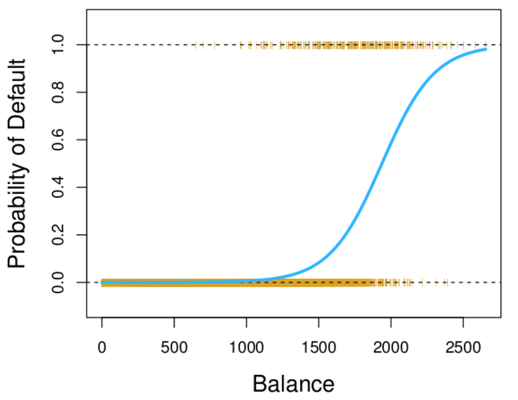
Try it out:
desmos.com/calculator/cw1pyzzqci

Logistic Regression

$$\Pr(\texttt{default} = \texttt{yes} \mid \texttt{balance}) = \frac{e^{\beta_0 + \beta_1 \texttt{balance}}}{1 + e^{\beta_0 + \beta_1 \texttt{balance}}}$$



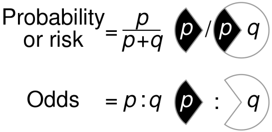
What will the drawn logistic regression classifer predict for each of the following values of `Balance`



Balance	Prediction
0	
500	
1000	
1500	
2000	
2500	

Odds

$$\frac{p(x)}{1-p(x)} = \frac{\Pr(Y=1 \mid X=x)}{1-\Pr(Y=1 \mid X=x)} = \frac{\Pr(Y=1 \mid X=x)}{\Pr(Y=0 \mid X=x)}$$



Examples:

- If the probability of default is 90% what are the odds?
  - $p(x) = 0.9$
  - $\frac{0.9}{1-0.9} = 9$
- If the odds are $1/3$, what is the probability of default?
  - $\frac{p}{1-p} = 1/3$
  - $3p = 1 - p$
  - $4p = 1$
  - $p = 1/4$

How to get logistic function

Assume the (natural) log odds (logits) follow a linear model

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

Solve for $p(x)$:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

Playing with the logistic function: desmos.com/calculator/cw1pyzzqci

Using coefficients to make predictions

	Coefficient	Std. error	z-statistic	p-value
`Intercept`	-10.6513	0.3612	-29.5	<0.0001
`balance`	0.0055	0.0002	24.9	<0.0001

What is the estimated probability of default for someone with a balance of \$1,000?

What is the estimated probability of default for someone with a balance of \$2,000:

Interpreting the coefficients

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

	Coefficient	Std. error	z-statistic	p-value
`Intercept`	-10.6513	0.3612	-29.5	<0.0001
`balance`	0.0055	0.0002	24.9	<0.0001

Confusion Matrix: Predicting `default` from `balance`

		True default status		
		No	Yes	Total
Predicted	No	9644	252	9896
default status	Yes	23	81	104
	Total	9667	333	10000

		True		
		Yes	No	Total
Predicted	Yes	a	b	$a+b$
	No	c	d	$c+d$
	Total	$a+c$	$b+d$	N

Do coding in jupyter notebook

Next time

Lec #	Date			Reading	HW
1	Mon	8/26	Intro / First day stuff / Python Review Pt 1	1	
2	Wed	8/28	What is statistical learning?	2.1	
3	Wed	9/4	Assessing Model Accuracy	2.2.1, 2.2.2	
4	Fri	9/6	Linear Regression	3.1	HW #1 Due Sun 9/8
5	Mon	9/9	More Linear Regression	3.1	
6	Wed	9/11	Multi-linear regression	3.2	
7	Fri	9/13	Probably more linear regression	3.3	Hw #2 Due Dun 9/15
8	Mon	9/16	Last of the linear regression		
9	Wed	9/18	Intro to classification, Bayes classifier, KNN classifier	2.2.3	
10	Fri	9/20	Logistic Regression	4.1, 4.2, 4.3.1-3	Hw #3 Due Sun 9/22
11	Mon	9/23	Multiple Logistic Regression / Multinomial Logistic Regression	4.3.4-5	
	Wed	9/25	***Project Day & Review***		
	Fri	9/27	**Midterm #1**		