ORIE 3120: Practical Tools for OR, DS, and ML

Logistic Regression (Lecture 13)

Professor Frazier

2021

Outline

Logistic Regression Model

Logistic Regression in Python

Maximum Likelihood Estimation

- Maximum Likelihood Estimation in Logistic Regression
- Maximum Likelihood Estimation in Linear Regression
- Maximum Likelihood Estimation for Uber offer acceptance

Logistic Regression with Multiple Trials

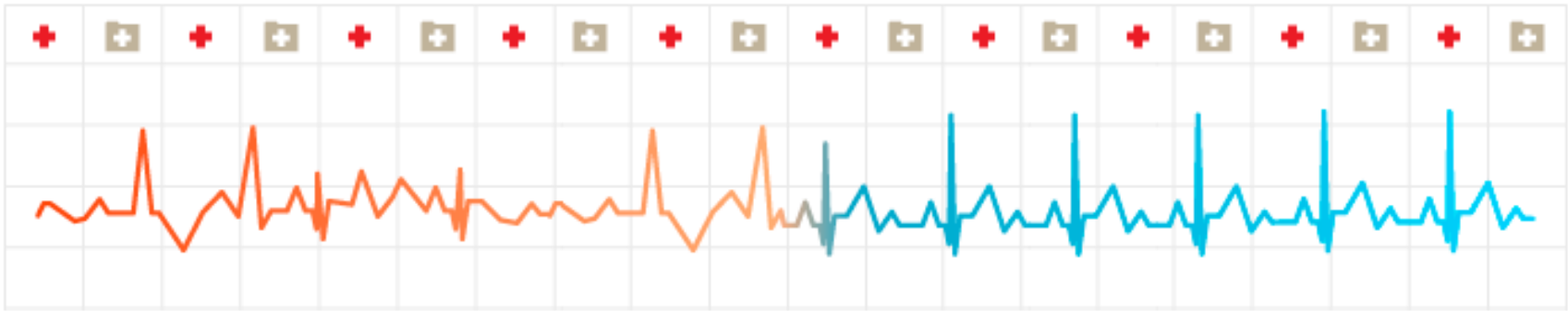
- Example: ingots data

Generalized Linear Models

Logistic Regression For Binary outcomes

- Logistic regression handles binary outcomes, e.g.,
  - "no" or "yes"
  - "defective" or "good"
  - "dead" or "alive"
  - often coded "0" or "1"
- We will model the conditional probability of "yes" given the predictors

Logistic Regression is Useful



Improve Healthcare, Win $3,000,000.

Heritage Health Prize

Identify patients who will be admitted to a hospital within the next year, using historical claims data.

Ends **12 months**

916 teams

$3 million

Data

For the ith case:

- $X_{i,1}, \ldots, X_{i,p}$ are the predictors
- $p(X_{i,1}, \ldots, X_{i,p})$ is the conditional probability of a "yes" or, equivalently, that $Y_i = 1$
- $Y_i | X_{i,1}, \ldots, X_{i,p}$ is Bernoulli$\{p(X_{i,1}, \ldots, X_{i,p})\}$
- $Pr(Y_i = y | X_{i,1}, \ldots, X_{i,p}) =$
  $p(X_{i,1}, \ldots, X_{i,p})^y \{1 - p(X_{i,1}, \ldots, X_{i,p})\}^{1-y}$

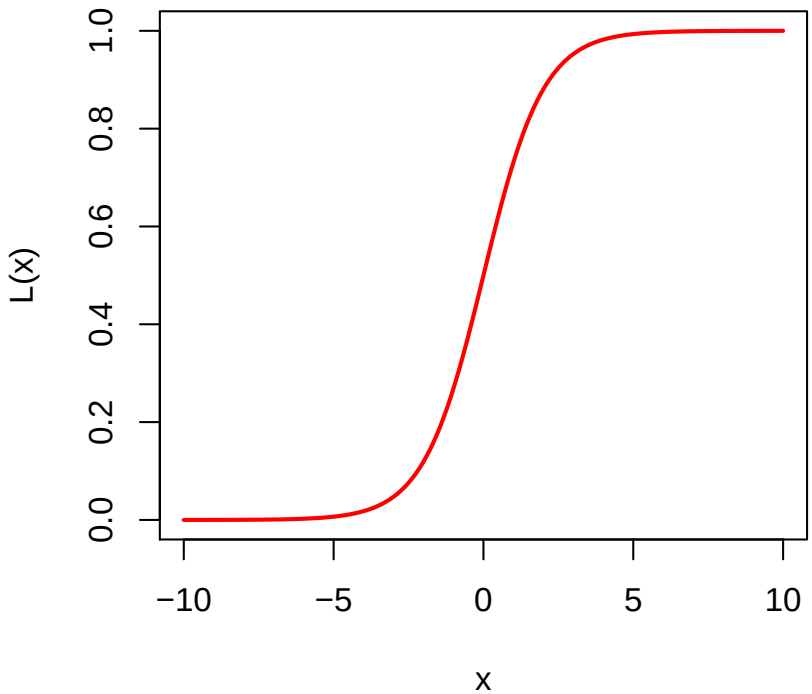
Problem modeling $p(X_1, \ldots, X_p)$ using a linear model

Recall: $p(X_1, \ldots, X_p)$ is the conditional probability of a "yes"

Linear model: $p(X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$

Problem: $\beta_0 + \beta_1 X_1 + \cdots \beta_p X_p$ can be above 1 or below 0

Logistic function



$$L(x) = \frac{1}{1 + \exp(-x)}$$

Logistic regression model

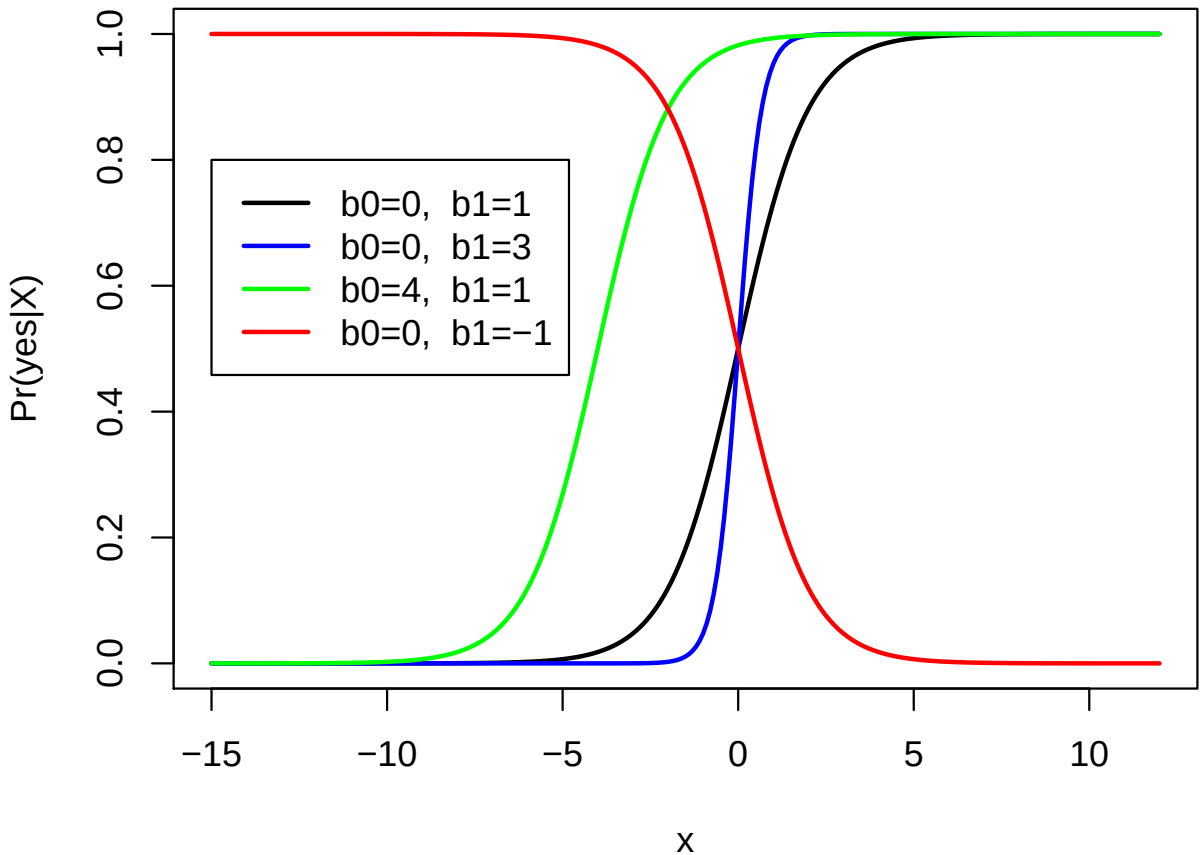
Logistic regression models $p(X_1, \ldots, X_n)$ using the logistic function:

$$p(X_1, \ldots, X_n) = L(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)$$

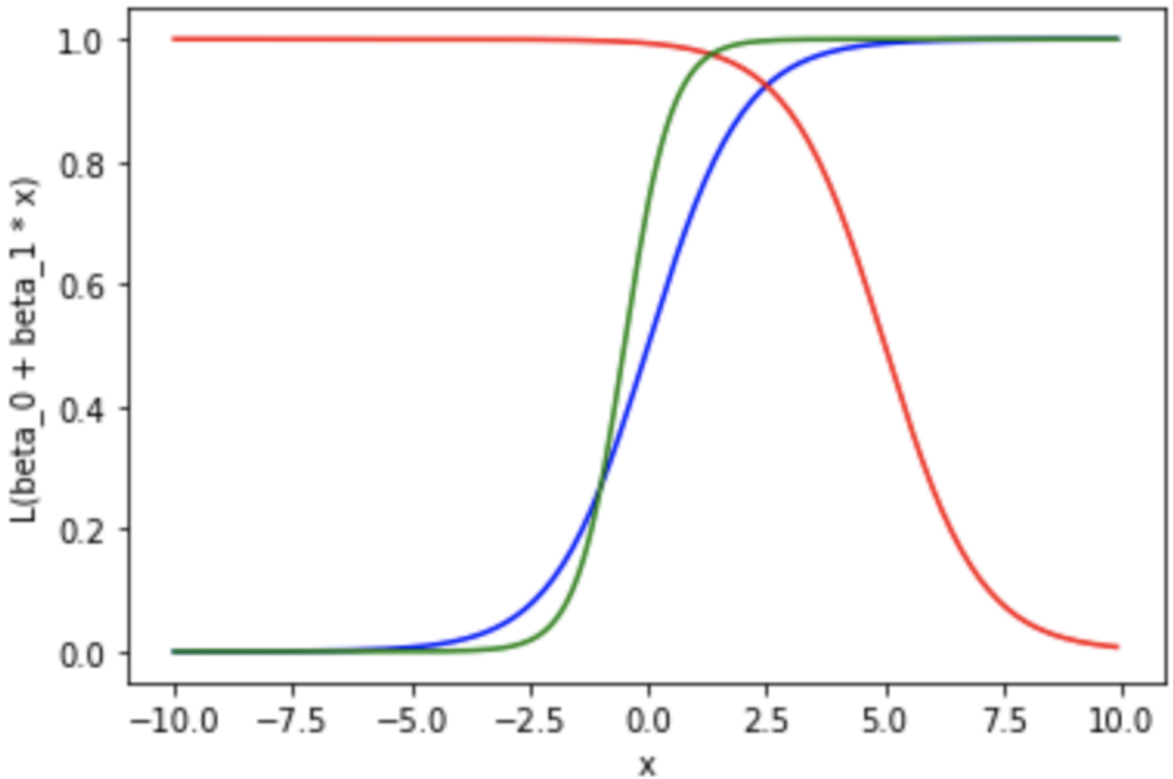
Let's look at the simplest case, $p = 1$:

$$p(X) = L(\beta_0 + \beta_1 X)$$

Some logistic models with one X



Questions 1-3



Among these functions, $L(x)$, $L(5-x)$, $L(1+2x)$, $L(5+x)$, we plotted three. Which function created each line?

Logit function

$$p(X_1, \ldots, X_n) = L(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)$$

implies that

$$L^{-1}\{p(X_1, \ldots, X_n)\} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

L^{-1} is called the "logit" function and is

$$L^{-1}(p) = \log\left(\frac{p}{1-p}\right)$$

Also called "log-odds"

Logit function

The "odds" for "yes" against "no" is

$$\frac{p}{1-p}$$

So the logistic model says that the log-odds equals

$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$

The logit function is called the "link" function because it links

- $p(X_1, \ldots, X_n)$, and
- $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$

Outline

Logistic Regression Model

Logistic Regression in Python

Maximum Likelihood Estimation

- Maximum Likelihood Estimation in Logistic Regression
- Maximum Likelihood Estimation in Linear Regression
- Maximum Likelihood Estimation for Uber offer acceptance

Logistic Regression with Multiple Trials

- Example: ingots data

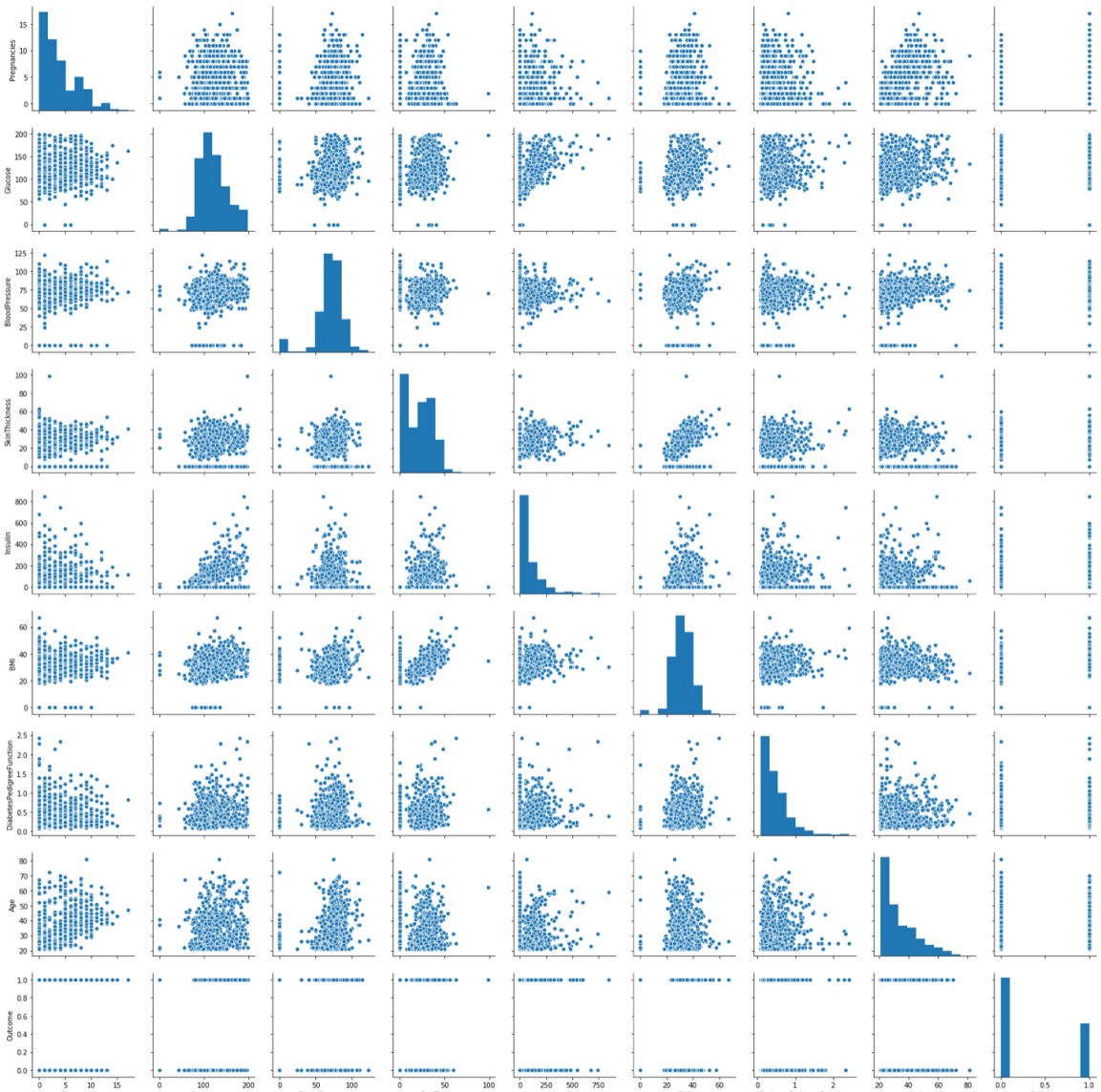
Generalized Linear Models

Logistic regression demo: Diabetes dataset

```
diabetes = pd.read_csv('diabetes.csv')
diabetes.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Logistic regression demo: Diabetes dataset



Fitting logistic regression

```
import statsmodels.api as sm
y = diabetes['Outcome']
features = diabetes.columns[:-1] # All columns except y
X = sm.add_constant(diabetes[features])
X.head()
```

	const	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	1.0	6	148	72	35	0	33.6	0.627	50
1	1.0	1	85	66	29	0	26.6	0.351	31
2	1.0	8	183	64	0	0	23.3	0.672	32
3	1.0	1	89	66	23	94	28.1	0.167	21
4	1.0	0	137	40	35	168	43.1	2.288	33

Fitting logistic regression

```
model = sm.Logit(y,X).fit()
model.summary()
```

Logit Regression Results

Dep. Variable:	Outcome	**No. Observations:**	768
Model:	Logit	**Df Residuals:**	759
Method:	MLE	**Df Model:**	8
Date:	Mon, 05 Apr 2021	**Pseudo R-squ.:**	0.2718
Time:	12:12:04	**Log-Likelihood:**	-361.72
converged:	True	**LL-Null:**	-496.74
Covariance Type:	nonrobust	**LLR p-value:**	9.652e-54

	coef	std err	z	P>|z|	[0.025	0.975]
const	-8.4047	0.717	-11.728	0.000	-9.809	-7.000
Pregnancies	0.1232	0.032	3.840	0.000	0.060	0.186
Glucose	0.0352	0.004	9.481	0.000	0.028	0.042
BloodPressure	-0.0133	0.005	-2.540	0.011	-0.024	-0.003
SkinThickness	0.0006	0.007	0.090	0.929	-0.013	0.014
Insulin	-0.0012	0.001	-1.322	0.186	-0.003	0.001
BMI	0.0897	0.015	5.945	0.000	0.060	0.119
DiabetesPedigreeFunction	0.9452	0.299	3.160	0.002	0.359	1.531
Age	0.0149	0.009	1.593	0.111	-0.003	0.033

Outline

Logistic Regression Model

Logistic Regression in Python

Maximum Likelihood Estimation

- Maximum Likelihood Estimation in Logistic Regression
- Maximum Likelihood Estimation in Linear Regression
- Maximum Likelihood Estimation for Uber offer acceptance

Logistic Regression with Multiple Trials

- Example: ingots data

Generalized Linear Models

Maximum Likelihood Estimation

Let y_i be the value of Y_i actually observed. Then the likelihood function evaluated at $\beta_0, \beta_1, \ldots, \beta_p$ is

$$\text{Likelihood}(\beta_0, \beta_1, \ldots, \beta_p) := Pr(Y_1 = y_1, \ldots, Y_n = y_n) =$$

$$= \prod_{i=1}^{n} p(X_{i,1}, \ldots, X_{i,p})^{y_i} \{1 - p(X_{i,1}, \ldots, X_{i,p})\}^{1-y_i}$$

Maximum likelihood estimation

- The maximum likelihood estimates are the values of $\beta_0, \beta_1, \ldots, \beta_p$ that make $\mathrm{Likelihood}(\beta_0, \beta_1, \ldots, \beta_p)$ as large as possible.
- The MLE's are computed by an optimization algorithm (often Newton's method)
- Details on computing the MLE are in more advanced courses (ORIE 4741)

Maximum likelihood is a general estimation method

- As we have seen, MLE is used for logistic regression
  - but MLE is not a special-purpose tool used just for logistic regression
- MLE = least squares for linear regression with normally distributed noise
- MLE is used for a wide variety of other statistical models
- MLE is, by far, the most popular estimation method

Minimizing the sum of squares is the same as maximum likelihood estimaton

- Linear regression assumed:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p} + \epsilon_i$$

  where ϵ_i are independent $\mathrm{Normal}(0, \sigma^2)$.
- This implies that Y_i is normal with mean $\beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}$ and variance σ^2
- A $\mathrm{Normal}(\mu, \sigma^2)$ distribution has density

$$\varphi(y; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

Question 4

What is the likelihood of the data in linear regression, $\text{Likelihood}(\beta_0, \ldots, \beta_p)$?

$$A. \prod_{i=1}^{n} \varphi(Y_i; \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}, \sigma^2)$$

$$B. \sum_{i=1}^{n} \varphi(Y_i; \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}, \sigma^2)$$

$$C. \prod_{i=1}^{n} p(X_{i,1}, \ldots, X_{i,p})^{Y_i} \{1 - p(X_{i,1}, \ldots, X_{i,p})\}^{1-Y_i}$$

$$D. \sum_{i=1}^{n} p(X_{i,1}, \ldots, X_{i,p})^{Y_i} \{1 - p(X_{i,1}, \ldots, X_{i,p})\}^{1-Y_i}$$

Minimizing the sum of squares is the same as maximum likelihood estimaton

- Let's suppose that $\hat{\beta} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$ is the maximum likelihood estimator, i.e., it maximizes $\mathrm{Likelihood}(\hat{\beta})$.
- We'll see that this gives us the same estimate as minimizing the sum of squared residuals.

$$
\begin{aligned}
\text{Likelihood}(\hat{\beta}) &= \prod_{i=1}^{n} \varphi(Y_i; \hat{\beta}_0 + \hat{\beta}_1 X_{i,1} + \cdots + \hat{\beta}_p X_{i,p}, \sigma^2) \\
&= \prod_{i=1}^{n} \varphi(Y_i; \hat{Y}_i, \sigma^2) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y_i - \hat{Y}_i)^2}{2\sigma^2}\right) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\hat{\epsilon_i}^2}{2\sigma^2}\right) \\
&= (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \hat{\epsilon_i}^2\right)
\end{aligned}
$$

- Applying a strictly increasing function to an objective function doesn't change the point that maximizes it.
- We are assuming $\hat{\beta}$ maximizes $\text{Likelihood}(\hat{\beta})$
- So $\hat{\beta}$ also maximizes $\log(\text{Likelihood}(\hat{\beta}))$
- $\hat{\beta}$ also minimizes $-\log(\text{Likelihood}(\hat{\beta}))$
- From a previous slide, $\text{Likelihood}(\hat{\beta}) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n} \hat{\epsilon_i}^2\right)$
- $-\log(\text{Likelihood}(\hat{\beta})) = -\log((2\pi\sigma^2)^{-n/2}) + \frac{1}{2\sigma^2}\sum_{i=1}^{n} \hat{\epsilon_i}^2$
- So $\hat{\beta}$ minimizes $\frac{1}{2\sigma^2}\sum_{i=1}^{n} \hat{\epsilon_i}^2$
- So $\hat{\beta}$ minimizes $\frac{1}{2\sigma^2}\sum_{i=1}^{n} \hat{\epsilon_i}^2$
- And $\hat{\beta}$ minimizes $\sum_{i=1}^{n} \hat{\epsilon_i}^2$
- This is the same as the estimator we used in linear regression!

MLE example: Uber offer acceptance

When a trip is offered to an Uber driver, (s)he either

- (A) accepts;
- (R) rejects; or
- (E) lets the offer expire.

We'll focus our data analysis on offers made in a particular area of a particular city in a particular time of week. E.g., the airport center in San Francisco, 5-6pm weekdays.

MLE example: Uber offer acceptance

Our model is that drivers' decisions (in SF city center 5-6pm weekdays) are independent across offers and satisfy

- $P(\textit{accept}) = p_A$
- $P(\textit{reject}) = p_R$
- $P(\textit{expire}) = 1 - p_A - p_R$

Let's estimate p_A and p_R from data.

Question 5

Our data shows that

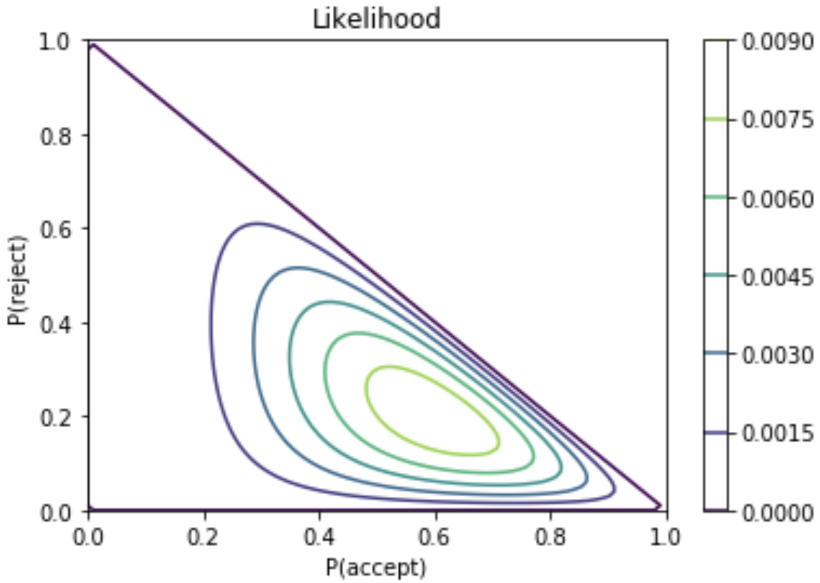
- offer 1 was accepted
- offer 2 was rejected
- offer 3 was accepted
- offer 4 expired
- offer 5 was accepted

(the number of offers would be much larger than 4, but let's keep it simple)

What is the likelihood?

1. $p_A\, p_R\, (1 - p_A - p_R)$
2. $p_A^3\, p_R\, (1 - p_A - p_R)$
3. $p_A\, p_R^3\, (1 - p_A - p_R)$
4. $p_A\, p_R\, (1 - p_A - p_R)^3$
5. $p_A^3\, p_R^3\, (1 - p_A - p_R)^3$

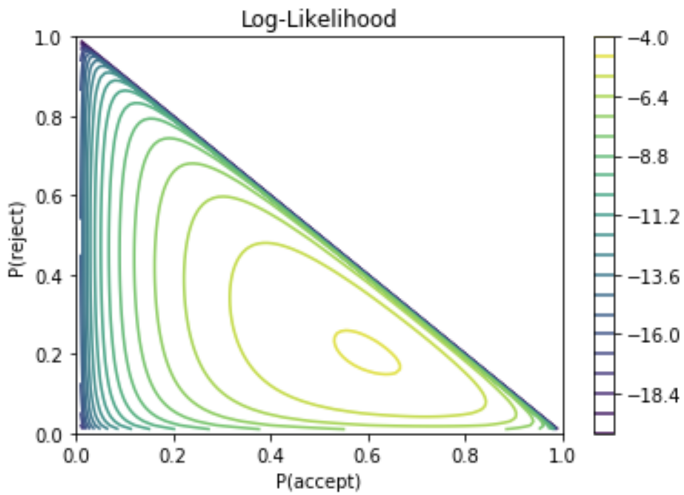
Question 6



What is the MLE?

1. $p_A = 0.33,\ p_R = 0.33$
2. $p_A = 0.2,\ p_R = 0.4$
3. $p_A = 0.4,\ p_R = 0.2$
4. $p_A = 0.6,\ p_R = 0.2$
5. $p_A = 0.2,\ p_R = 0.6$

MLE example: Uber offer acceptance



The log(likelihood) is maximized at the same p_A and p_R.

This example is called *multinomial logistic regression* and it's useful when we want to model people choosing among more than 2 options.

Outline

Logistic Regression Model

Logistic Regression in Python

Maximum Likelihood Estimation

- Maximum Likelihood Estimation in Logistic Regression
- Maximum Likelihood Estimation in Linear Regression
- Maximum Likelihood Estimation for Uber offer acceptance

Logistic Regression with Multiple Trials

- Example: ingots data

Generalized Linear Models

Logistic Regression with Multiple Trials

- Sometimes we have multiple observations (trials) at the same sets of predictors
- You could make each observation its own datapoint and use the method for fitting logistic regression we already learned
- BUT, if there are a lot of trials, this can make the data big and fitting slow
- INSTEAD, we'll show you how to generalize logistic regression to directly handle data with multiple trials

Logistic Regression with Multiple Trials

For the ith case:

- $X_{i,1}, \ldots, X_{i,p}$ are the predictors
- n_i is the number of "trials"
- $p(X_{i,1}, \ldots, X_{i,p})$ is the conditional probability of a "yes" or, equivalently, that $Y_i = 1$
- $Y_i | X_{i,1}, \ldots, X_{i,p}$ is Binomial$\{p(X_{i,1}, \ldots, X_{i,p}), n_i\}$
- So

$$Pr(Y_i = y | X_{i,1}, \ldots, X_{i,p})$$

$$= \binom{n_i}{y} p(X_{i,1}, \ldots, X_{i,p})^y \{1 - p(X_{i,1}, \ldots, X_{i,p})\}^{n_i - y}$$

for $y = 0, \ldots, n_i$

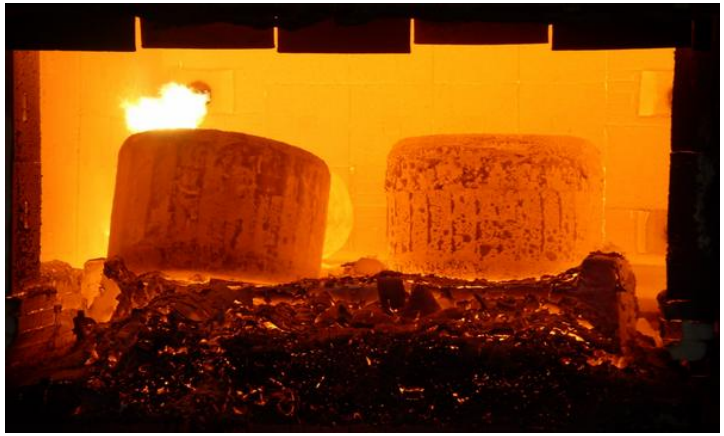
We Still Use Maximum Likelihood Estimation

Let y_i be the value of Y_i actually observed. Then the likelihood function evaluated at $\beta_0, \beta_1, \ldots, \beta_p$ is

$$\text{Likelihood}(\beta_0, \beta_1, \ldots, \beta_p) := Pr(Y_1 = y_1, \ldots, Y_n = y_n) =$$

$$= \prod_{i=1}^{n} \binom{n_i}{y_i} p(X_{i,1}, \ldots, X_{i,p})^{y_i} \{1 - p(X_{i,1}, \ldots, X_{i,p})\}^{n_i - y_i}$$

Example: Heating steel ingots to be rolled is hard!



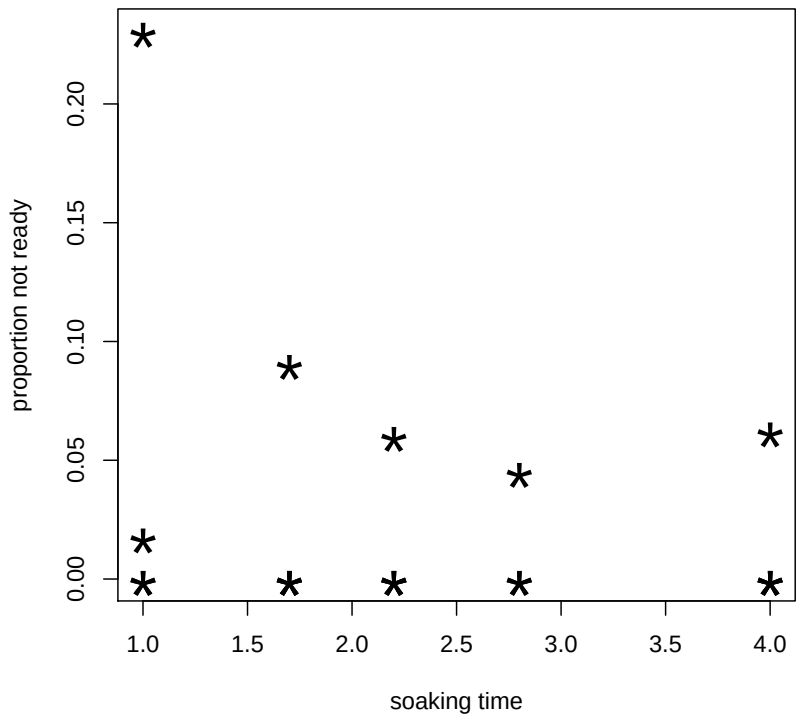
Example: ingots data

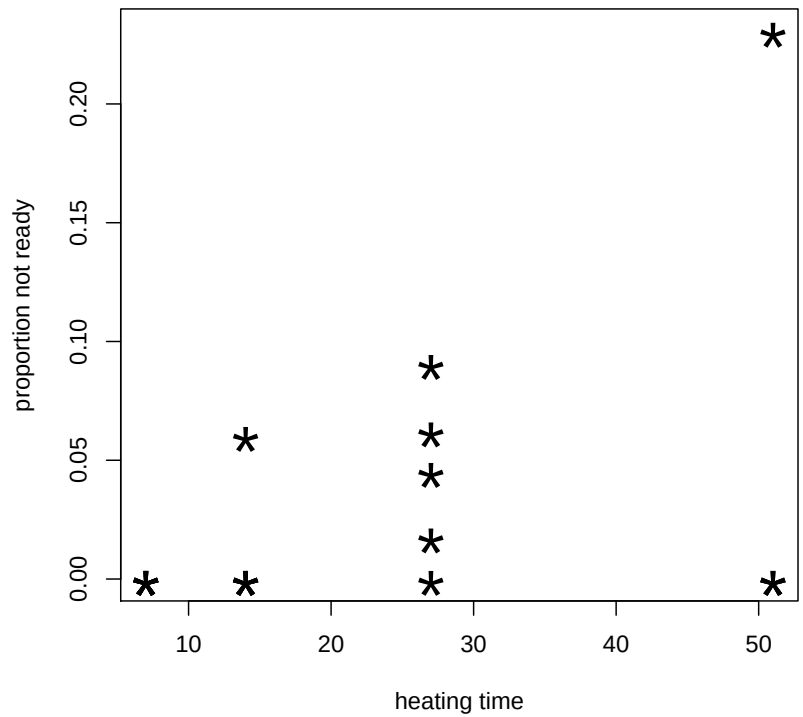
Soak Time	Heat Time	Not Ready	n_i
1	7	0	10
1	14	0	31
1	27	1	56
1	51	3	13
1.7	7	0	17
1.7	14	0	43
1.7	27	4	44
1.7	51	0	1
2.2	7	0	7
2.2	14	2	33
2.2	27	0	21
2.2	51	0	1
2.8	7	0	12
2.8	14	0	31
2.8	27	1	22
2.8	51	0	0
4	7	0	9
4	14	0	19
4	27	1	16
4	51	0	1

n_i = number of ingots prepared

proportion not ready = (Not Ready) $/ n_i$

Let's look at the data





How to fit logistic regression with multiple trials

Transform the outcome data to have success and failure

```
ingot['failure'] = ingot['not_ready']
ingot['success'] = ingot['n'] - ingot['not_ready']
y = ingot[['success', 'failure']]
X = sm.add_constant(ingot[['heat', 'soak']])
```

Use statsmodels.GLM

```
model = sm.GLM(y, X, family=sm.families.Binomial(), \
               intercept=True).fit()
model.summary()
```

How to fit logistic regression with multiple trials

Dep. Variable:	['success', 'failure']	**No. Observations:**	19
Model:	GLM	**Df Residuals:**	16
Model Family:	Binomial	**Df Model:**	2
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-14.040
Date:	Mon, 05 Apr 2021	**Deviance:**	13.753
Time:	10:59:47	**Pearson chi2:**	13.5
No. Iterations:	7		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	5.5592	1.120	4.965	0.000	3.365	7.754
heat	-0.0820	0.024	-3.456	0.001	-0.129	-0.036
soak	-0.0568	0.331	-0.171	0.864	-0.706	0.592

Outline

Logistic Regression Model

Logistic Regression in Python

Maximum Likelihood Estimation

- Maximum Likelihood Estimation in Logistic Regression
- Maximum Likelihood Estimation in Linear Regression
- Maximum Likelihood Estimation for Uber offer acceptance

Logistic Regression with Multiple Trials

- Example: ingots data

Generalized Linear Models

Generalized Linear Models

Logistic regression is an example of a generalized linear model (GLM)

A GLM is similar to a LM, except

- the linear prediction equation

$$E(Y|X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  is replaced by

$$E(Y|X_1, \ldots, X_p) = H(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)$$

  for a suitable function H
- $H =$ logistic function for logistic regression

Generalized Linear Models, cont.

- The conditional normal distribution of Y given $X_1, \ldots, X_p$ is replaced by another family of distributions
  - binomial distributions for logistic regression
- Poisson regression is another example of a GLM
  - Y_i is Poisson
  - $H(x) = \exp(x)$ because the mean of a Poisson is positive

GLMs

$$E(Y|X_1, \ldots, X_p) = H(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)$$

implies that

$$\frac{\partial}{\partial X_j} E(Y|X_1, \ldots, X_p) = H'(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)\beta_j$$

so the coefficients in a GLM can be interpreted in roughly the same way

in a LM

Logistic regression with multiple trials is a GLM

In logistic regression with multiple trials,
$Y_i \mid X_{i,1}, \ldots, X_{i,p} \sim \text{Binomial}(p(X_{i,1}, \ldots, X_{i,p}), n_i)$

This is what `family` means:

```
model = sm.GLM(y, X, family=sm.families.Binomial(), \
                    intercept=True).fit()
```

The argument `intercept=True` only influences one of the model-quality statistics produced and can be ignored.