Logistic Regression

Prof Wells

STA 295: Stat Learning

April 4th, 2024

Outline

- Discuss logistic regression for classification
- Describe extensions of logistic regression: multivariate and multinomial
- Implement logistic regression in R

Section 1

Logistic Regression

Classificaiton Problems

- Suppose Y is a categorical variable with levels $A_1, A_2, \ldots, A_k$.

Classificaiton Problems

- Suppose Y is a categorical variable with levels $A_1, A_2, \ldots, A_k$.
  - Example: Let Y indicate whether it is raining in Portland at noon on 10/25/21.
  - Levels: $A_1 = \text{Raining}$, $A_2 = \text{Not Raining}$.

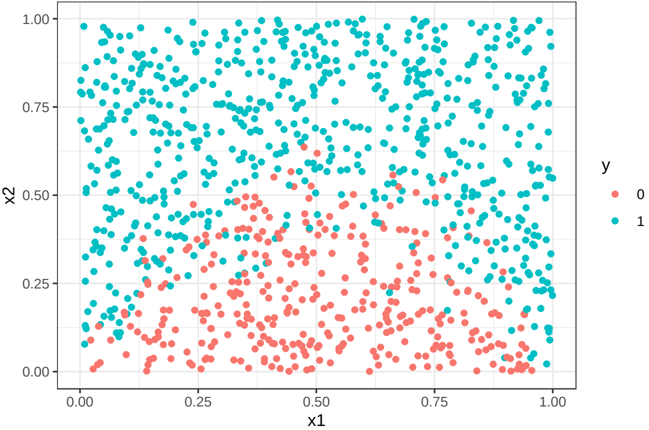
Classificaiton Problems

- Suppose Y is a categorical variable with levels $A_1, A_2, \ldots, A_k$.
  - Example: Let Y indicate whether it is raining in Portland at noon on 10/25/21.
  - Levels: $A_1 = \text{Raining}$, $A_2 = \text{Not Raining}$.
- Goal: Build a model f to classify an observation into levels $A_1, A_2, \ldots, A_k$ based on the values of several predictors $X_1, X_2, \ldots, X_p$ (quantitative or categorical)

$$\hat{Y} = f(X_1, X_2, \ldots, X_p) \qquad \text{where } f \text{ take values in } \{A_1, \ldots, A_k\}$$

Classification Regions

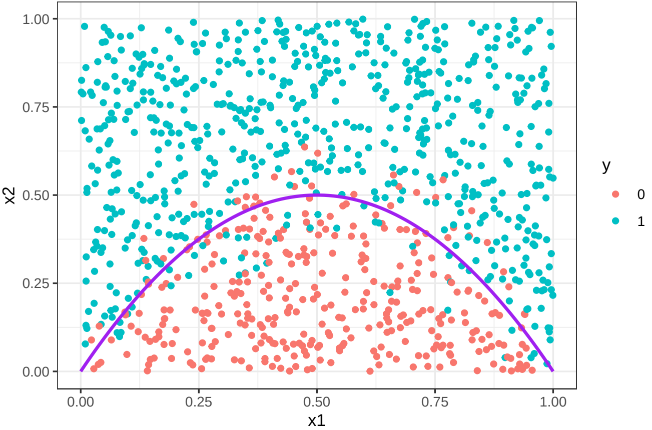
Any classification model will divide predictor space into unions of regions, where each point in a region will be classified in the same way.



Different models will have different geometries for classification boundaries.

Classification Regions

Any classification model will divide predictor space into unions of regions, where each point in a region will be classified in the same way.



The purple line indicates the optimal decision boundary.

The Bayes Classifier and KNN

- The Bayes classifier theoretically minimizes error rate

$$f(x_0) = \text{argmax}_{A_j} P(Y = A_j \mid X = x_0)$$

The Bayes Classifier and KNN

- The Bayes classifier theoretically minimizes error rate

$$f(x_0) = \mathrm{argmax}_{A_j} P(Y = A_j \mid X = x_0)$$

  - In practice, these conditional probabilities are not known.

The Bayes Classifier and KNN

- The Bayes classifier theoretically minimizes error rate

$$f(x_0) = \mathrm{argmax}_{A_j} P(Y = A_j \mid X = x_0)$$

  - In practice, these conditional probabilities are not known.
- But we can approximate them using *KNN*:

$$P(Y = A_j \mid X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$$

The Bayes Classifier and KNN

- The Bayes classifier theoretically minimizes error rate

$$f(x_0) = \mathrm{argmax}_{A_j} P(Y = A_j \mid X = x_0)$$

  - In practice, these conditional probabilities are not known.

- But we can approximate them using *KNN*:

$$P(Y = A_j \mid X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$$

- Our model for P is therefore $\hat{P}_j(x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$.

The Bayes Classifier and KNN

- The Bayes classifier theoretically minimizes error rate

$$f(x_0) = \mathrm{argmax}_{A_j} P(Y = A_j \mid X = x_0)$$

  - In practice, these conditional probabilities are not known.
- But we can approximate them using *KNN*:

$$P(Y = A_j \mid X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$$

- Our model for P is therefore $\hat{P}_j(x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$.
- And our classifier model is $\hat{g}(x_0) = \mathrm{argmax}_{A_j} \hat{P}_j(x_0)$

Why not always just use KNN?

1. KNN has very low training time (basically none), but often large test time (especially for large K)

Why not always just use KNN?

1. KNN has very low training time (basically none), but often large test time (especially for large K)
2. KNN models are hard to interpret, so often not ideal for inference questions.

Why not always just use KNN?

1. KNN has very low training time (basically none), but often large test time (especially for large K)
2. KNN models are hard to interpret, so often not ideal for inference questions.
3. If a linear or more structured model is more appropriate (i.e. accurately captures the true form of f), then KNN will be less stable.

Why not always just use KNN?

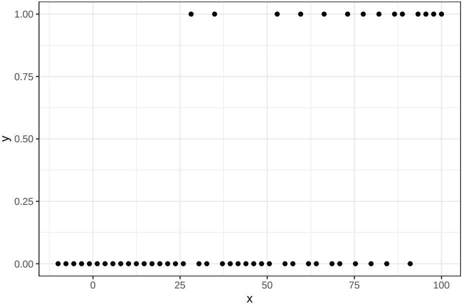
1. KNN has very low training time (basically none), but often large test time (especially for large K)
2. KNN models are hard to interpret, so often not ideal for inference questions.
3. If a linear or more structured model is more appropriate (i.e. accurately captures the true form of f), then KNN will be less stable.
4. KNN suffers from the "curse of dimensionality". For fixed K and large p, adding more predictors increases bias and variance.

Why not always just use KNN?

1. KNN has very low training time (basically none), but often large test time (especially for large K)
2. KNN models are hard to interpret, so often not ideal for inference questions.
3. If a linear or more structured model is more appropriate (i.e. accurately captures the true form of f), then KNN will be less stable.
4. KNN suffers from the "curse of dimensionality". For fixed K and large p, adding more predictors increases bias and variance.
5. KNN requires large sample sizes (compared to alternatives)

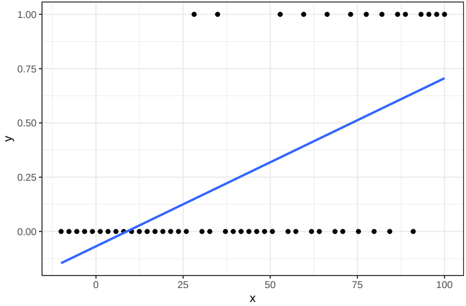
Alternatives

- Suppose Y is a binary categorical variable with a single quantitative predictor X. We want to model $p(X) = P(Y = 1|X)$



Alternatives

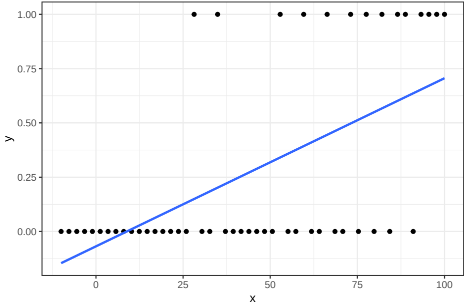
- Suppose Y is a binary categorical variable with a single quantitative predictor X. We want to model $p(X) = P(Y = 1|X)$



- Linear model: $p(X) = \beta_0 + \beta_1 X = -0.07 + 0.008X$

Alternatives

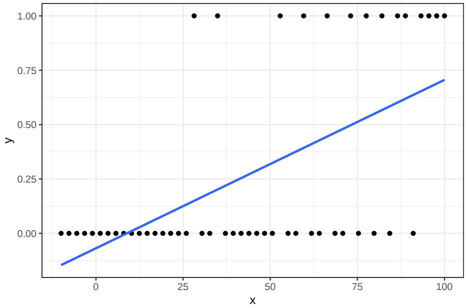
- Suppose Y is a binary categorical variable with a single quantitative predictor X. We want to model $p(X) = P(Y = 1|X)$



- Linear model: $p(X) = \beta_0 + \beta_1 X = -0.07 + 0.008X$
- Predict 1 if $\hat{P}(x) \geq 0.5$, and 0 otherwise.

Alternatives

- Suppose Y is a binary categorical variable with a single quantitative predictor X. We want to model $p(X) = P(Y = 1|X)$



- Linear model: $p(X) = \beta_0 + \beta_1 X = -0.07 + 0.008X$
- Predict 1 if $\hat{P}(x) \geq 0.5$, and 0 otherwise.
  - Solving the linear equation, predict 1 if $X \geq 73.4$

Problems with linear model

1. Our prediction $p(X)$ may take values outside 0 and 1.

Problems with linear model

1. Our prediction $p(X)$ may take values outside 0 and 1.
2. Too inflexible (enormous bias).

Problems with linear model

1. Our prediction $p(X)$ may take values outside 0 and 1.
2. Too inflexible (enormous bias).
3. In practice, $p(X)$ is rarely close to linear.

Odds

- Suppose a certain event occurs with probability p. The odds of the event occurring are

$$\text{odds} = \frac{p}{1-p}$$

Odds

- Suppose a certain event occurs with probability p. The odds of the event occurring are

$$\text{odds} = \frac{p}{1-p}$$

  - If $p = .75$, then $\text{odds} = 3$ (or 3 to 1).
  - If $p = .5$, then $\text{odds} = 1$ (or even odds).

Odds

- Suppose a certain event occurs with probability p. The odds of the event occurring are

$$\text{odds} = \frac{p}{1-p}$$

  - If $p = .75$, then $\text{odds} = 3$ (or 3 to 1).
  - If $p = .5$, then $\text{odds} = 1$ (or even odds).
- But odds compress unlikely events towards 0, while stretching likely events towards infinity.

Odds

- Suppose a certain event occurs with probability p. The odds of the event occurring are

$$\text{odds} = \frac{p}{1-p}$$

  - If $p = .75$, then $\text{odds} = 3$ (or 3 to 1).
  - If $p = .5$, then $\text{odds} = 1$ (or even odds).
- But odds compress unlikely events towards 0, while stretching likely events towards infinity.
  - Events that are less likely to happen than not have odds between 0 and 1, while events that are more likely to happen than not have odds between 1 and infinity.

Odds

- Suppose a certain event occurs with probability p. The odds of the event occurring are

$$\text{odds} = \frac{p}{1-p}$$

  - If $p = .75$, then $\text{odds} = 3$ (or 3 to 1).
  - If $p = .5$, then $\text{odds} = 1$ (or even odds).
- But odds compress unlikely events towards 0, while stretching likely events towards infinity.
  - Events that are less likely to happen than not have odds between 0 and 1, while events that are more likely to happen than not have odds between 1 and infinity.
- So instead, we consider log odds:

$$\text{log odds} = \ln \frac{p}{1-p} = \ln p - \ln(1-p)$$

Logistic Regression

- Suppose Y is binary categorical, and that the log odds of the event "$Y = 1$" is linear in X. That is,

Logistic Regression

- Suppose Y is binary categorical, and that the log odds of the event "$Y = 1$" is linear in X. That is,

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X$$

Logistic Regression

- Suppose Y is binary categorical, and that the log odds of the event "$Y = 1$" is linear in X. That is,

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X$$

- Increasing X by 1 increases the log odds of $Y = 1$ by a constant amount.

Logistic Regression

- Suppose Y is binary categorical, and that the log odds of the event "$Y = 1$" is linear in X. That is,

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X$$

- Increasing X by 1 increases the log odds of $Y = 1$ by a constant amount.
- Increasing X by 1 increases the odds of $Y = 1$ by a constant *relative rate*

Logistic Regression

- Suppose Y is binary categorical, and that the log odds of the event "$Y = 1$" is linear in X. That is,

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X$$

- Increasing X by 1 increases the log odds of $Y = 1$ by a constant amount.
- Increasing X by 1 increases the odds of $Y = 1$ by a constant *relative rate*
- Solving for odds:

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

Logistic Regression

- Suppose Y is binary categorical, and that the log odds of the event "$Y = 1$" is linear in X. That is,

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X$$

- Increasing X by 1 increases the log odds of $Y = 1$ by a constant amount.
- Increasing X by 1 increases the odds of $Y = 1$ by a constant *relative rate*
- Solving for odds:

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

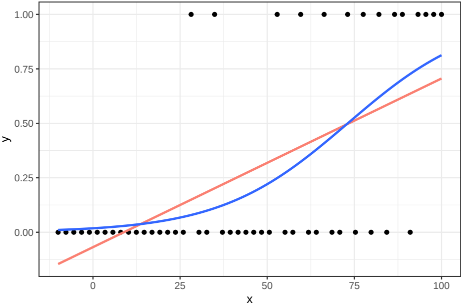
- Solving for $p(X)$:

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

The Logistic Curve

- The conditional probability $p(X)$ takes the form of a logistic curve:

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

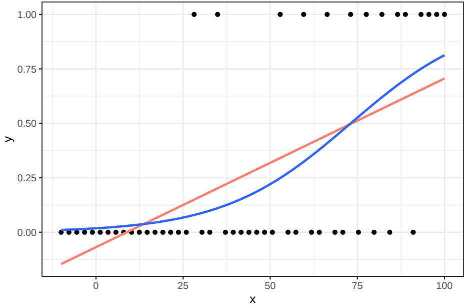


- Logistic model: $p(X) = \frac{e^{-4+0.05X}}{1+e^{-4+0.05X}}$

The Logistic Curve

- The conditional probability $p(X)$ takes the form of a logistic curve:

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

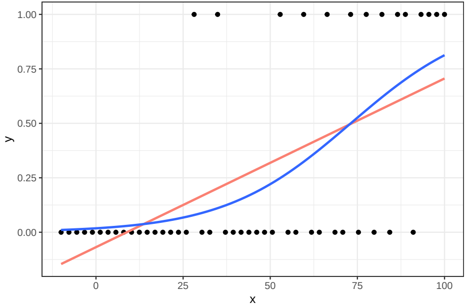


- Logistic model: $p(X) = \frac{e^{-4+0.05X}}{1+e^{-4+0.05X}}$
- Predict 1 if $\hat{P}(x) \geq 0.5$ (or if $\log \text{odds} \geq 0$)

The Logistic Curve

- The conditional probability $p(X)$ takes the form of a logistic curve:

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



- Logistic model: $p(X) = \frac{e^{-4+0.05X}}{1+e^{-4+0.05X}}$
- Predict 1 if $\hat{P}(x) \geq 0.5$ (or if $\log \text{odds} \geq 0$)
  - Solving the linear equation, predict 1 if $X \geq 73.1$

Multiple Logistic Regression

- Nothing stops us from modeling Y based on more than 1 predictor.

Multiple Logistic Regression

- Nothing stops us from modeling Y based on more than 1 predictor.

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

Multiple Logistic Regression

- Nothing stops us from modeling Y based on more than 1 predictor.

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

- Solving for $p(X)$:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$

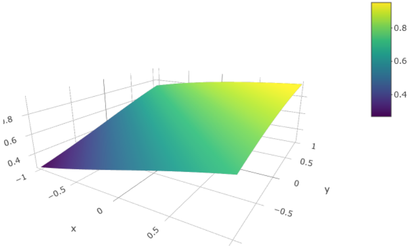
Multiple Logistic Regression

- Nothing stops us from modeling Y based on more than 1 predictor.

$$\ln \frac{p(X)}{1-p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

- Solving for $p(X)$:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$



Applications of Logistic Regression

Logistic Regression is the most commonly used binary classification method. . .

Applications of Logistic Regression

Logistic Regression is the most commonly used binary classification method...

1. For historical reasons
2. Due to its relative simplicity
3. For ease of interpretation
4. Because it often gives reasonable predictions

Applications of Logistic Regression

Logistic Regression is the most commonly used binary classification method. . .

1. For historical reasons
2. Due to its relative simplicity
3. For ease of interpretation
4. Because it often gives reasonable predictions

Logistic regression has been used to. . .

1. Create spam filters
2. Forecast election results
3. Investigate health outcomes based on patient risk factors

Section 2

Interpreting and Estimating Coefficients

Effect of Coefficients in Logistic Model

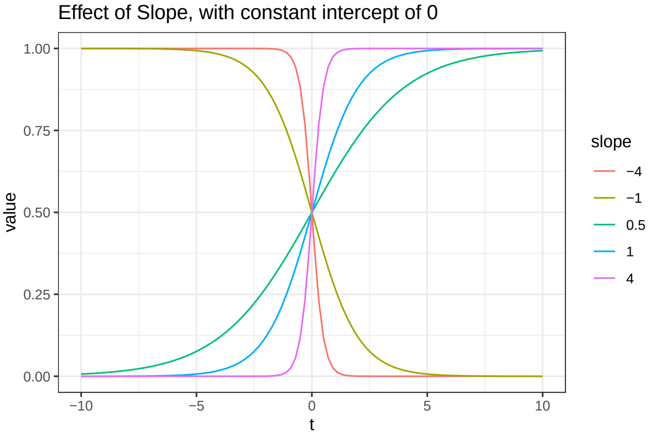
Consider a logistic regression model for a binary variable Y based on predictor X.

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

Effect of Coefficients in Logistic Model

Consider a logistic regression model for a binary variable Y based on predictor X.

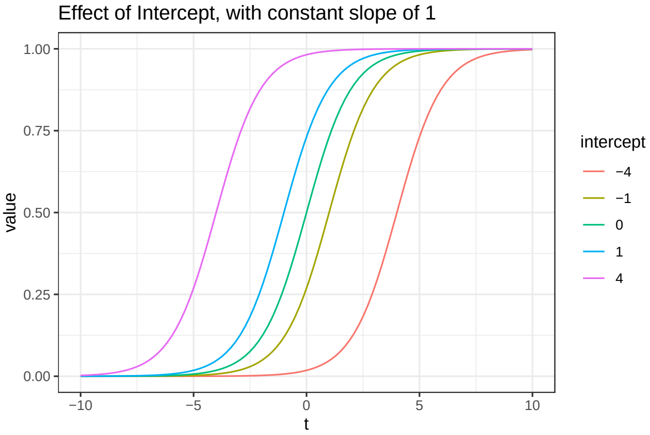
$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X \qquad p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$



Effect of Coefficients in Logistic Model

Consider a logistic regression model for a binary variable Y based on predictor X.

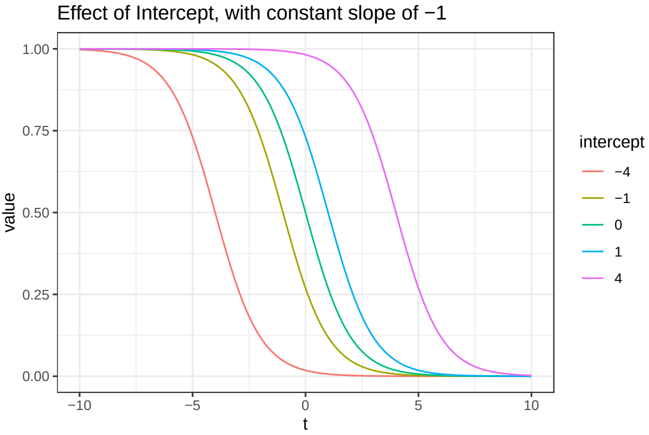
$$\ln \frac{p(X)}{1-p(X)} = \beta_0 + \beta_1 X \qquad p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$



Effect of Coefficients in Logistic Model

Consider a logistic regression model for a binary variable Y based on predictor X.

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



Interpreting Coefficients

Consider a logistic regression model for a binary variable Y based on predictor X.

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

Interpreting Coefficients

Consider a logistic regression model for a binary variable Y based on predictor X.

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

- The intercept β_0 is the log-odds when $X = 0$. Alternatively,

$$\text{odds}(Y=1|X=0) = e^{\beta_0} \qquad \text{Prob}(Y=1|X=0) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$$

Interpreting Coefficients

Consider a logistic regression model for a binary variable Y based on predictor X.

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

- The intercept β_0 is the log-odds when $X = 0$. Alternatively,

$$\text{odds}(Y = 1 | X = 0) = e^{\beta_0} \qquad \text{Prob}(Y = 1 | X = 0) = \frac{e^{\beta_0}}{1 + e^{\beta_0}}$$

- The slope β_1 is rate of change in log-odds when X increases by 1 unit. Alternatively,

Interpreting Coefficients

Consider a logistic regression model for a binary variable Y based on predictor X.

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

- The intercept β_0 is the log-odds when $X = 0$. Alternatively,

$$\text{odds}(Y=1|X=0) = e^{\beta_0} \qquad \text{Prob}(Y=1|X=0) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$$

- The slope β_1 is rate of change in log-odds when X increases by 1 unit. Alternatively,

$$\begin{aligned} \text{odds}(Y=1|X=x+1) = e^{\beta_0+\beta_1(x+1)} &= e^{\beta_0+\beta_1 x+\beta_1} = e^{\beta_1} \cdot e^{\beta_0+\beta_1 x} \\ &= e^{\beta_1} \cdot \text{odds}(Y=1|X=x) \end{aligned}$$

which shows that when X increases by 1 unit, the odds change by a factor of e^{β_1}.

Regression Coefficient Estimates

- Assume that the log-odds of $Y = 1$ is indeed linear in $X_1, \ldots, X_p$, so that

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

Regression Coefficient Estimates

- Assume that the log-odds of $Y = 1$ is indeed linear in $X_1, \ldots, X_p$, so that

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  - We need to estimate the parameters $\beta_0, \beta_1, \ldots, \beta_p$ based on training data.

Regression Coefficient Estimates

- Assume that the log-odds of $Y = 1$ is indeed linear in $X_1, \ldots, X_p$, so that

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  - We need to estimate the parameters $\beta_0, \beta_1, \ldots, \beta_p$ based on training data.
- We could use the Method of Least Squares, as we did with Linear Regression.

$$\beta = (X^T X)^{-1} X^T y$$

Regression Coefficient Estimates

- Assume that the log-odds of $Y = 1$ is indeed linear in $X_1, \dots, X_p$, so that

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  - We need to estimate the parameters $\beta_0, \beta_1, \dots, \beta_p$ based on training data.
- We could use the Method of Least Squares, as we did with Linear Regression.

$$\beta = (X^T X)^{-1} X^T y$$

  - But this won't necessarily produce accurate estimates, since residuals tend not to be approximately Normally distributed

Regression Coefficient Estimates

- Assume that the log-odds of $Y = 1$ is indeed linear in $X_1, \ldots, X_p$, so that

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  - We need to estimate the parameters $\beta_0, \beta_1, \ldots, \beta_p$ based on training data.

- We could use the Method of Least Squares, as we did with Linear Regression.

$$\beta = (X^T X)^{-1} X^T y$$

  - But this won't necessarily produce accurate estimates, since residuals tend not to be approximately Normally distributed

- Instead, we use the method of **Maximum Likelihood** (ML)

Regression Coefficient Estimates

- Assume that the log-odds of $Y = 1$ is indeed linear in $X_1, \ldots, X_p$, so that

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  - We need to estimate the parameters $\beta_0, \beta_1, \ldots, \beta_p$ based on training data.
- We could use the Method of Least Squares, as we did with Linear Regression.

$$\beta = (X^T X)^{-1} X^T y$$

  - But this won't necessarily produce accurate estimates, since residuals tend not to be approximately Normally distributed
- Instead, we use the method of **Maximum Likelihood** (ML)
  - We consider all possible values of $\beta_0, \ldots, \beta_p$, and choose the ones for which the observed data x had highest probability of occurring.

Regression Coefficient Estimates

- Assume that the log-odds of $Y = 1$ is indeed linear in $X_1, \ldots, X_p$, so that

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  - We need to estimate the parameters $\beta_0, \beta_1, \ldots, \beta_p$ based on training data.
- We could use the Method of Least Squares, as we did with Linear Regression.

$$\beta = (X^T X)^{-1} X^T y$$

  - But this won't necessarily produce accurate estimates, since residuals tend not to be approximately Normally distributed
- Instead, we use the method of **Maximum Likelihood** (ML)
  - We consider all possible values of $\beta_0, \ldots, \beta_p$, and choose the ones for which the observed data x had highest probability of occurring.
  - I.e. we choose the model which is most consistent with the data.

Regression Coefficient Estimates

- Assume that the log-odds of $Y = 1$ is indeed linear in $X_1, \ldots, X_p$, so that

$$\ln \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

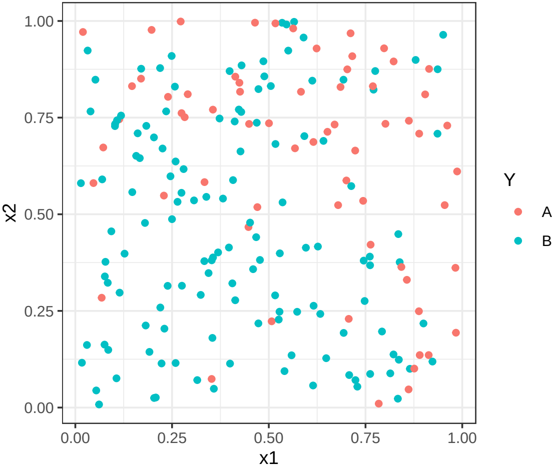
  - We need to estimate the parameters $\beta_0, \beta_1, \ldots, \beta_p$ based on training data.
- We could use the Method of Least Squares, as we did with Linear Regression.

$$\beta = (X^T X)^{-1} X^T y$$

  - But this won't necessarily produce accurate estimates, since residuals tend not to be approximately Normally distributed
- Instead, we use the method of **Maximum Likelihood** (ML)
  - We consider all possible values of $\beta_0, \ldots, \beta_p$, and choose the ones for which the observed data x had highest probability of occurring.
  - I.e. we choose the model which is most consistent with the data.
  - How? Use numeric methods to optimize (and R)

Logistic Regression in R

Recall the simulation of 200 points from the model $p = \frac{x_1^2 + x_2^2}{2}$:



Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
## Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

- Logistic regression requires the response to either be binary numeric (0 or 1) or a binary **factor**

Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

- Logistic regression requires the response to either be binary numeric (0 or 1) or a binary **factor**
    - The model will estimate the probability of the second level (i.e. $P(Y = B)$)

Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

- Logistic regression requires the response to either be binary numeric (0 or 1) or a binary **factor**
  - The model will estimate the probability of the second level (i.e. $P(Y = B)$)
- To change this, we can either recode the response as numeric:

```
sim_data$Y <- ifelse(sim_data$Y == "A", 1, 0)
head(sim_data$Y)
```

```
## [1] 1 0 0 0 0 0
```

Logistic Regression in R

Before we fit the model, we need to pay attention to the response variable:

```
str(sim_data$Y)
```

```
##  Factor w/ 2 levels "A","B": 1 2 2 2 2 2 1 1 2 2 ...
```

- Logistic regression requires the response to either be binary numeric (0 or 1) or a binary **factor**
  - The model will estimate the probability of the second level (i.e. $P(Y = B)$)
- To change this, we can either recode the response as numeric:

```
sim_data$Y <- ifelse(sim_data$Y == "A", 1, 0)
head(sim_data$Y)
```

```
## [1] 1 0 0 0 0 0
```

- Or we can relevel the factor:

```
sim_data$Y <- factor(sim_data$Y, levels = c("B", "A"))
head(sim_data$Y)
```

```
## [1] A B B B B B
## Levels: B A
```

Logistic Regression in R

We fit a logistic regression model using the `glm` function.

Logistic Regression in R

We fit a logistic regression model using the `glm` function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

Logistic Regression in R

We fit a logistic regression model using the `glm` function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

- We need to include `family = "binomial"` to tell R we want **logistic** regression

Logistic Regression in R

We fit a logistic regression model using the glm function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

- We need to include `family = "binomial"` to tell R we want **logistic** regression
- We can view the fitted model using `summary`, or just the coefficient estimates using `$coefficients`

Logistic Regression in R

We fit a logistic regression model using the glm function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

- We need to include `family = "binomial"` to tell R we want **logistic** regression
- We can view the fitted model using `summary`, or just the coefficient estimates using `$coefficients`

```
summary(sim_logistic)$coefficients
```

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -3.472875  0.5685977 -6.107789 1.010206e-09
## x1           2.746111  0.6570948  4.179170 2.925746e-05
## x2           2.448198  0.5996131  4.082962 4.446520e-05
```

Logistic Regression in R

We fit a logistic regression model using the `glm` function.

```
sim_logistic <- glm(Y ~ x1 + x2, data = sim_data, family = "binomial")
```

- We need to include `family = "binomial"` to tell R we want **logistic** regression
- We can view the fitted model using `summary`, or just the coefficient estimates using `$coefficients`

```
summary(sim_logistic)$coefficients
```

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -3.472875  0.5685977 -6.107789 1.010206e-09
## x1           2.746111  0.6570948  4.179170 2.925746e-05
## x2           2.448198  0.5996131  4.082962 4.446520e-05
```

- From the table, our logistic regression model is

$$\log \frac{p(X_1, X_2)}{1 + p(X_1, X_2)} = -3.47 + 2.75 \cdot X_1 + 2.45 \cdot X_2$$

Classification

To classify using logistic regression, we set a classification threshhold (usually 0.5) and predict $Y = 1$ if $P(x) > 0.5$.

Classification

To classify using logistic regression, we set a classification threshhold (usually 0.5) and predict $Y = 1$ if $P(x) > 0.5$.

- Note that if $P(x) = 0.5$, then odds are $P(x)/(1 - P(x)) = 0.5/0.5 = 1$, and the log odds are $\log(1) = 0$.

Classification

To classify using logistic regression, we set a classification threshhold (usually 0.5) and predict $Y = 1$ if $P(x) > 0.5$.

- Note that if $P(x) = 0.5$, then odds are $P(x)/(1 - P(x)) = 0.5/0.5 = 1$, and the log odds are $\log(1) = 0$.
  - Thus, we classify $Y = 1$ if $\log \mathrm{odds} > 0$.

Classification

To classify using logistic regression, we set a classification threshhold (usually 0.5) and predict $Y = 1$ if $P(x) > 0.5$.

- Note that if $P(x) = 0.5$, then odds are $P(x)/(1 - P(x)) = 0.5/0.5 = 1$, and the log odds are $\log(1) = 0$.
    - Thus, we classify $Y = 1$ if $\log \mathrm{odds} > 0$.
- Our fitted model predicting whether $Y = A$ was

$$\log \frac{p(X_1, X_2)}{1 + p(X_1, X_2)} = -3.47 + 2.75 \cdot X_1 + 2.45 \cdot X_2$$

and so we classify $Y = A$ if

$$0 < -3.47 + 2.75 \cdot X_1 + 2.45 \cdot X_2$$

or equivalently, if

$$X_2 > (3.47 - 2.75 \cdot X_1)/2.45$$

Decision Boundary

The logistic decision boundary is $X_2 = (3.47 - 2.75 \cdot X_1)/2.45$ (purple)

- We classify as A all points above this line, and classify as B all points below this line.
- The Bayes Classifier decision boundary shown in black