Logistic Regression
Probabilistic Classification Models

- **Generative Model**
  - Learns $\text{Pr}[C \mid X]$ indirectly
    - $\text{P}[C \mid X] = \text{P}[X, C] / \text{P}[X] = \text{P}[X \mid C] \text{P}[C] / \text{P}[X]$
    - Depends on prior knowledge: $\text{P}[C]$ is called the **prior**
    - Calculate the **likelihood** $\text{Pr}[X \mid C]$: probability of the features, given the labels
    - Finally, apply Bayes’ rule to calculate $\text{Pr}[C \mid X]$
      - Example: Naive Bayes

- **Discriminative Model**
  - Learns $\text{Pr}[C \mid X]$ directly
  - Example: Logistic Regression
Probabilistic Classification Models

- Learn $\Pr[C \mid X, \theta]$
  - Model parameters $\theta$ imply a probability distribution over classes $C$, given feature values $X$
  - Learn $\theta$ that minimizes error / maximizes accuracy

- Maximum *a posteriori* (MAP) principle
  - To classify, choose a class $C$ that maximizes $P[C \mid X, \theta]$
Probabilistic Classification Models

- Learn $\Pr[C \mid X, \theta]$
  - Model parameters $\theta$ imply a probability of class $C$, given feature values $X$
  - Learn $\theta$ that minimizes error / maximizes accuracy

- Maximum a posteriori (MAP) principle
  - To classify, choose a class $C$ that maximizes $\Pr[C \mid X, \theta]$

$$c_{\text{MAP}} \in \arg\max_{c \in C} P(C \mid X)$$
$$= \arg\max_{c \in C} \frac{P(X \mid C) P(C)}{P(X)}$$
$$\propto \arg\max_{c \in C} P(X \mid C) P(C)$$
Classification via Regression

- Regression for binary classification: yes/no, true/false, 1/0
- Here are data about whether or not students passed an exam as a function of on hours studied
- We want to use regression to predict the probability of passing.
Classification via Regression

- Here are data about whether or not students passed an exam as a function of hours studied.
- The data have binary classifications: yes/no, true/false, 1/0
- We want to use regression to predict the probability of success (passing).
- But we cannot use linear regression, because probabilities outside the range [0, 1] are nonsensical.
We can try non-linear regression

This is the **logistic** function

It is a gentle S-shaped curve, also called a **sigmoid**

Using it, we can contain probability values within a sensible range

\[
\frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}
\]
Logistic Regression

- We can solve a regression with the sigmoid to get something like this.
- The logistic function constrains predictions to fall between 0 and 1.
- These predictions can be interpreted as passing probabilities.
- At three hours there’s a ~60% chance of passing; at 1 hour there’s a ~10% chance; at 5 hours, ~95%.
Statistical Model

We seek a model of how binary response $Y$ varies with explanatory variable $X$ (or with multiple explanatory variables $X_1, \ldots, X_n$)

- We assume a Bernoulli response variables $Y$, with probability of success $p$
- The probability of success $p$ is a nonlinear function of the explanatory variables, via a sigmoidal:

\[ p = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \]
Logit Function

- Inverse of the logistic function
- The logistic function bounds the $Y$ values between 0 and 1
- The logit function bounds the $X$ values b/n 0 and 1. The $Y$ values are unbounded, so a linear model now makes sense.
- This function is the log of the odds of $p$, where the odds is the ratio of $p$ to $1-p$
  - If $p = \frac{1}{2}$, then the odds are 1:1 (even odds)
  - If $p = \frac{2}{3}$, then the odds are 2:1
Statistical Model

We seek a model of how binary response $Y$ varies with explanatory variable $X$ (or with multiple explanatory variables $X_1, \ldots, X_n$)

- We assume a Bernoulli response variables $Y$, with probability of success $p$
- The probability of success $p$ is a nonlinear function of the explanatory variables, via a sigmoidal:

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

- In other words, we assume the log of the odds of $p$ is linear: i.e., $\text{logit}(p) = \beta_0 + \beta_1 x$
What is the likelihood function?

\( F \) is the logistic function.

\[
P(y_i = 1 \mid x_i) = F(\beta_0 + \beta_1 x_i)
\]

\[
P(y_i = 0 \mid x_i) = 1 - F(\beta_0 + \beta_1 x_i)
\]

\[
L(y_i \mid x_i) = (F(\beta_0 + \beta_1 x_i))^{y_i} (1 - F(\beta_0 + \beta_1 x_i))^{1-y_i}
\]

\[
L(\{y_i\}_{i=1}^{n} \mid \{x_i\}_{i=1}^{n}) = \prod_{i=1}^{n} P(y_i \mid x_i)
\]
What is the log likelihood function?

\[
\log L(\{y_i\}_{i=1}^n \mid \{x_i\}_{i=1}^n) = \log \prod_{i=1}^n P(y_i \mid x_i)
\]

\[
= \sum_{i=1}^n \log P(y_i \mid x_i)
\]

\[
= \sum_{i=1}^n \log \left\{ (F(\beta_0 + \beta_1 X))^{y_i} (1 - F(\beta_0 + \beta_1 X))^{1-y_i} \right\}
\]

\[
= \sum_{i=1}^n y_i \log F(\beta_0 + \beta_1 X) + (1 - y_i) \log(1 - F(\beta_0 + \beta_1 X))
\]

Let’s proceed as usual. Take derivatives wrt \(\beta_0\) and \(\beta_1\). Set them equal to zero. Etc. UH OH! Nonlinear system of equations. No closed-form solution. Iterative methods.
We can apply a decision rule to transform our logistic regression output into classifications.

One rule is: round the probability.

Class 0 if $< 50\%$; class 1 if $\geq 50\%$.

This decision boundary is not the regression curve; it is just a line at an arbitrary cutoff probability.
Logistic Regression is a Linear Classifier

The decision boundary is the set of $x$ such that

$$\frac{1}{1 + e^{-\theta \cdot x}} = 0.5$$

A little bit of algebra shows that this is equivalent to

$$1 = e^{-\theta \cdot x}$$

and, taking the natural log of both sides,

$$0 = -\theta \cdot x = -\sum_{i=0}^{n} \theta_i x_i$$

so the decision boundary is linear. (A linear combination of the parameters and the feature values.)
Multiple logistic regression

- Just as in linear regression, we can assume multiple explanatory variables.
- If in addition to hours studied, we also recorded hours slept, we could build a 3D logistic regression model.
Logistic Regression in R
Logistic regression in R

- Similar to the `lm` function we can use `glm`

```r
> fit <- glm(pass ~ hours,
             data = class_stats,
             family = binomial(link = "logit"))
```

- Still have formula and data frame reference
- Must include `family = ...` to use logistic regression
  - `glm` is used for many other things; we’ll only use logistic regression
Logistic regression in R

- We can predict values using `predict`,

\[
> \text{preds} \leftarrow \text{predict(fit, newdata, type = "response")}
\]

- `type = "response"` ensures that we get a value from 0 to 1
- We can convert these to classes by using

\[
> \text{preds} \leftarrow \text{ifelse(preds < 0.5, 0, 1)}
\]