Data Mining
W4240 Section 001

Giovanni Motta

Columbia University, Department of Statistics

October 14, 2015

Outline

Outline

Classification

How do we answer the following questions?

- A 30 year old female arrives at the ER with chest pain and difficulty breathing. From a list of possible conditions, which one is she most likely to have?
- A bank has a usage history for a credit card, including transaction dates, amounts, locations, and merchant classification. Which transactions are fraudulent?
- Given a set of gene expression data for a specific tissue sample, can we tell whether that sample is normal or cancerous?

Classification

Setup:

- have a data set $(x_1, y_1), \ldots, (x_n, y_n)$
- the values for $y_1, \ldots, y_n$ are *categorical*
- want to fit a model to $(x_1, y_1), \ldots, (x_n, y_n)$ so that we can predict y_{new} from x_{new}

Errors are not really additive in this case:

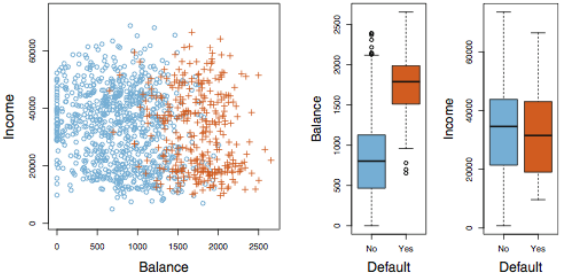
$$Y_i = \begin{cases} g_1 & \text{with probability } p(g_1 \,|\, X_i = x_i) \\ \vdots & \vdots \\ g_d & \text{with probability } p(g_d \,|\, X_i = x_i) \end{cases}$$

We want to find a function $\hat{f}(X)$ that solves

$$\min_f \mathbb{E}\left[\mathbf{1}_{\{Y \neq f(X)\}}\right] = \min_f \text{ prob. } f \text{ predicts wrong label}$$

Classification

Here is some credit card data from the `Default` dataset in `ISLR`. We would like to predict whether a holder will `default` based on `balance`, `income`, and `student` status.[1]



[1] Some images included from *An Introduction to Statistical Learning* by James, Witten, Hastie and Tibshirani.

Classification

So how do we solve a classification problem?

- kNN: find k nearest neighbors, pick a majority label
- I want something less flexible... what about linear models?

Idea: make categorical response numeric!

- set `default` equal to 1
- set `not default` equal to 0
- ... and then fit a linear regression model

Classification

Let's do this with the `Default` data:

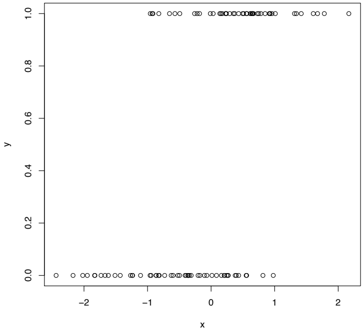
```
> library(ISLR)
> names(Default)
> Default[1:5,]
> default.dummy <- rep(0,length(Default$default))
> default.dummy[Default$default=="Yes"] <- 1
> df.default <- data.frame(default = default.dummy,student = Default$student,
 + balance= Default$balance, income = Default$income )
> fit.lm.default <- lm(default~.,data=df.default)
> fit.lm.default
Call:
lm(formula = default ~ ., data = df.default)

Coefficients:
(Intercept)   studentYes
 -8.118e-02   -1.033e-02
    balance       income
  1.327e-04    1.992e-07
```

Categorical Responses

Now let's make some data in R with a single covariate.

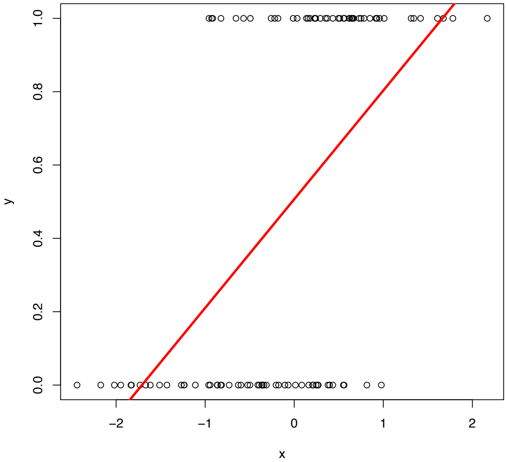
Use the function rbern() in package Rlab to get Bernoulli random variables (can also use rmultinom()):

```
> x <- rnorm(100)
> y <- rbern(100,exp(2*x)/(1+exp(2*x)))
> plot(x,y)
```



Categorical Responses

Fit a linear model to this data...



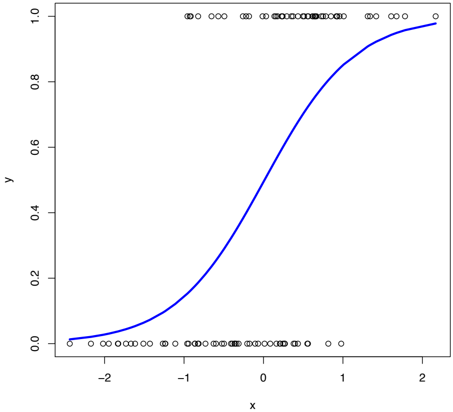
What issues exist here?

Outline

Categorical Responses

Idea: fit a *probability* of seeing $y_i = 1$ given $x_i = x$

- all output values should be between 0 and 1
- gives some notion of uncertainty

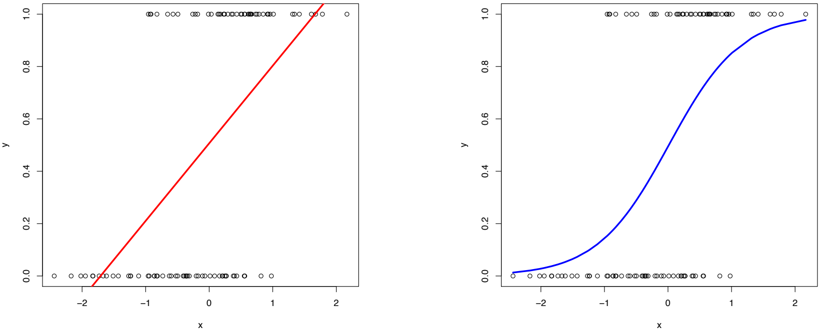


Why is this a sensible thing to do (vs linear regression)?

Notation: Classification and Logistic Regression

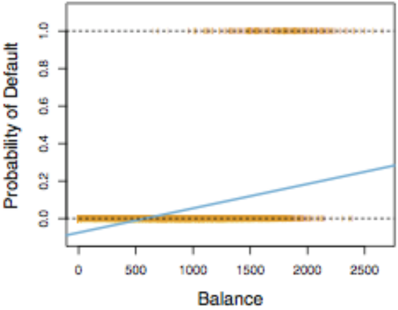
There is some notational complexity here:

- We are fitting categorial data
- Our goal is classification
- We will fit a probability curve $\pi(x) : \mathbb{R} \to [0,1]$
- So this is often called a *regression*.
- (we will shortly introduce a *logistic* curve).



Categorical Responses

We can do this with `Default` as well:



Logistic Regression

Model:

- We get observations:

$$Y|\pi(x) \sim Bernoulli\,(\pi(x))$$

- Want to model $\pi(x) : \mathbb{R} \to [0,1]$
- $\{0,1\}$ observations are noise atop this function (like ϵ_i)
- Log likelihood of $\pi(x)$:

$$\ell(\pi(x)) = y \log \pi(x) + (1-y)\log(1-\pi(x))$$

- What functions can we choose for $\pi(x)$?

Logistic Regression

Idea: use linear regression!

However,

- outputs for linear regression are in $(-\infty, \infty)$
- outputs for this are in $(0, 1)$

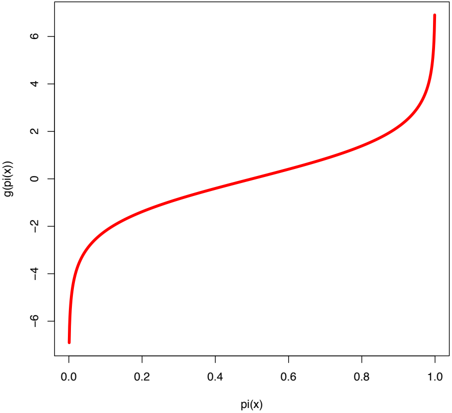
What if we map $(0, 1)$ to $(-\infty, \infty)$?
Enter the *logit* function.

Logistic Regression

Logit function:

$$g(x) = \log \frac{\pi(x)}{1 - \pi(x)} = \log(\pi(x)) - \log(1 - \pi(x))$$

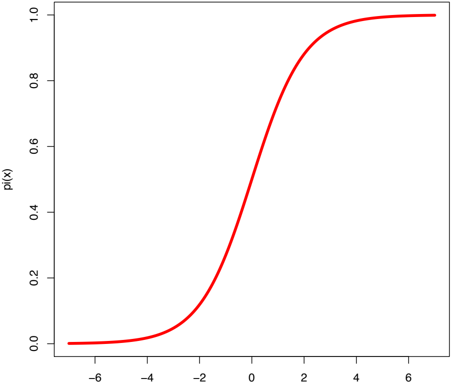
This is a log-odds function. $g(x) : [0, 1] \to \mathbb{R}$



Logistic Regression

Note:

- $g(x)$ maps probabilities to real numbers
- $\pi(x)$ maps real numbers to probabilities
- Write $\pi(x)$ as a function of $g(x)$
- Introducing the *logistic* function:



Logistic Regression

Where do we stand:

- Logit function:

$$g(x) = \log \frac{\pi(x)}{1-\pi(x)} = \log(\pi(x)) - \log(1-\pi(x))$$

- We can pick a function with $(-\infty, \infty)$ range for $g(x)$...
- ...and there is a corresponding $\pi(x)$ that behaves as we wish.
- Hence we call the logit function a *link* function.
- Ideas?
- We will model the log odds as **linear** in the covariates:

$$\log \frac{\pi(x)}{1-\pi(x)} = g(x) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

Outline

Logistic Regression

We will model the log odds as **linear** in the covariates:

$$\log \frac{\pi(x)}{1-\pi(x)} = g(x) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

What does this mean?

- the *odds* is the ratio between heads and tails: $\frac{\pi(x)}{1-\pi(x)}$
- linear assumption: a one unit increase in x_j produces a β_j unit increase in $\log \frac{\pi(x)}{1-\pi(x)}$
- linear assumption: (equivalent) a one unit increase in x_j multiplies the odds by $\underline{e^{\beta_j}}$
- note: rate of change in $\pi(x)$ per unit of X depends on X!

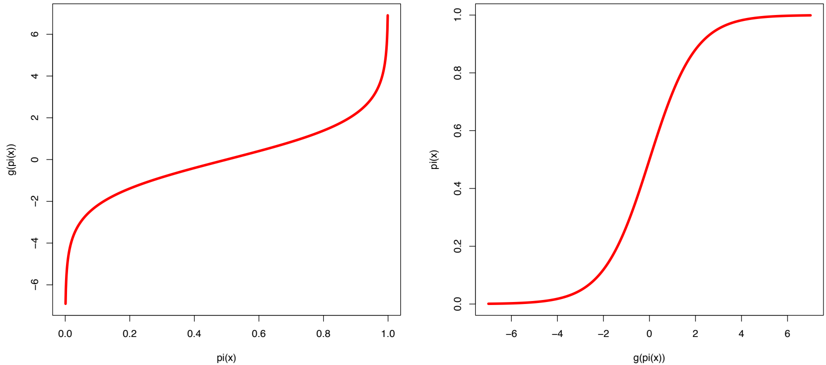
Logistic Regression

$\pi(x)$ is sometimes easier to think about than $\log \frac{\pi(x)}{1-\pi(x)} = g(x)$

To get from $\log \frac{\pi(x)}{1-\pi(x)}$ to $\pi(x)$, we use the *logistic function*,

$$\begin{aligned}
\pi(x) &= \frac{e^{g(x)}}{1+e^{g(x)}} \\
&= \frac{1}{e^{-g(x)}+1} \\
&= \frac{e^{\beta_0+\beta_1 x_1+\ldots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\ldots+\beta_p x_p}} \\
&= \frac{1}{e^{-\beta_0-\beta_1 x_1-\ldots-\beta_p x_p}+1}
\end{aligned}$$

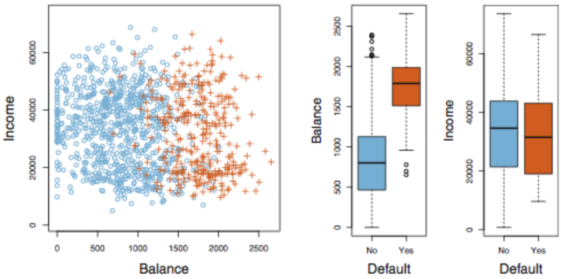
Logistic Regression



Logit (left) and logistic (right) functions

Classification

Let's return to this credit card data from the `Default` dataset in `ISLR`. We would like to predict whether a holder will `default` based on `balance`, `income`, and `student` status.



Logistic Regression

$$\hat{g}(x) = \hat{\beta}_0 + \hat{\beta}_1 X$$

$$\hat{\pi}(x) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}}$$

$$\begin{aligned}
\hat{\pi}(\texttt{default} = \texttt{Yes} | \texttt{student} = \texttt{Yes}) &= \frac{e^{-3.5041 + 0.4049 \times 1}}{1 + e^{-3.5041 + 0.4049 \times 1}} = 0.0431 \\
\hat{\pi}(\texttt{default} = \texttt{Yes} | \texttt{student} = \texttt{No}) &= \frac{e^{-3.5041 + 0.4049 \times 0}}{1 + e^{-3.5041 + 0.4049 \times 0}} = 0.0292
\end{aligned}$$

Logistic Regression

Let's try to understand what these coefficients mean.

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−3.5041	0.0707	−49.55	<0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

How does student status affect default probability?

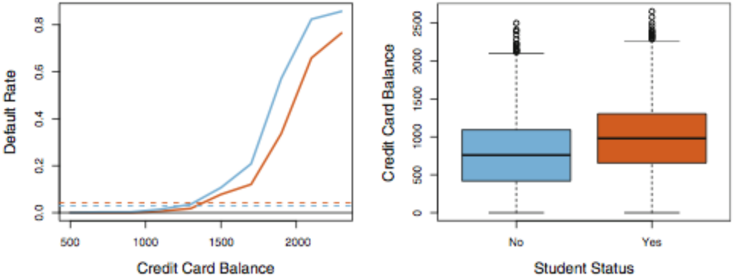
```
> mean(df.default$default[df.default$student=="Yes"])
[1] 0.04313859
> mean(df.default$default[df.default$student=="No"])
[1] 0.02919501
```

What does the logistic regression say?

Logistic Regression

Let's try to understand what these coefficients mean.

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.8690	0.4923	−22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	−0.6468	0.2362	−2.74	0.0062



Outline

Logistic Regression

Model:

$$\pi(x_i)\,|\,x_i = \frac{e^{\beta_0+\sum_{j=1}^{p}\beta_j x_{i,j}}}{1+e^{\beta_0+\sum_{j=1}^{p}\beta_j x_{i,j}}}$$

$$y_i\,|\,\pi(x_i) \sim Bernoulli(\pi(x_i))$$

So how do we find $\hat{\beta}$?

Use maximum likelihood

$$\ell(\beta) = \sum_{i=1}^{n} y_i \log\left(\frac{e^{\beta_0+\sum_{j=1}^{p}\beta_j x_{i,j}}}{1+e^{\beta_0+\sum_{j=1}^{p}\beta_j x_{i,j}}}\right) + (1-y_i)\log\left(\frac{1}{1+e^{\beta_0+\sum_{j=1}^{p}\beta_j x_{i,j}}}\right)$$

$$\hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} y_i \log\left(\frac{e^{\beta_0+\sum_{j=1}^{p}\beta_j x_{i,j}}}{1+e^{\beta_0+\sum_{j=1}^{p}\beta_j x_{i,j}}}\right) + (1-y_i)\log\left(\frac{1}{1+e^{\beta_0+\sum_{j=1}^{p}\beta_j x_{i,j}}}\right)$$

Logistic Regression

Set

$$p(x_i; \beta) = \frac{e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}{1 + e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}$$

Then, simplify

$$\begin{aligned}
\ell(\beta) &= \sum_{i=1}^{n} \{y_i \log\, p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta))\} \\
&= \sum_{i=1}^{n} \left\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right\}
\end{aligned}$$

As usual, take the derivate and set equal to 0

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{n} x_i (y_i - p(x_i; \beta)) = 0$$

($p + 1$ equations that are *non-linear* in β)

Logistic Regression

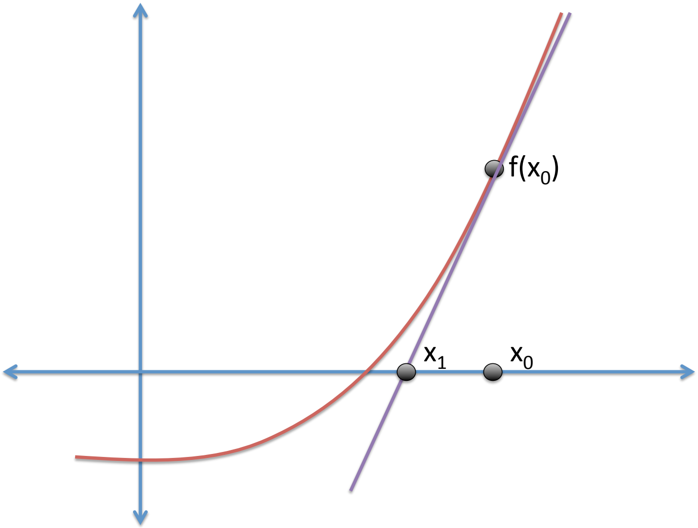
How do we find a solution?

- problem is still convex...
- use Newton-Raphson method

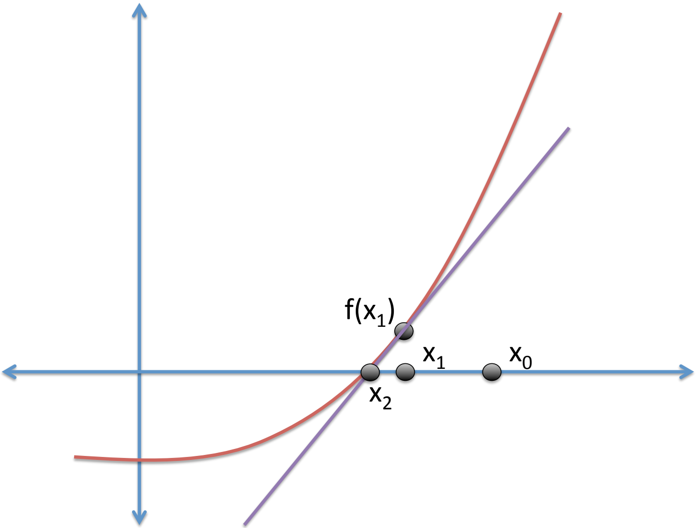
General idea:

- want to find x such that $f(x) = 0$
- start at point x_0
- approximate root by $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
- repeat approximation

Newton-Raphson Method



Newton-Raphson Method



Logistic Regression

Want to find roots for the $p+1$ equations

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{n} x_i(y_i - p(x_i;\beta)) = 0$$

We need to take the second derivative! (Find the Hessian)

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{n} x_i x_i^T p(x_i;\beta)(1 - p(x_i;\beta))$$

Update:

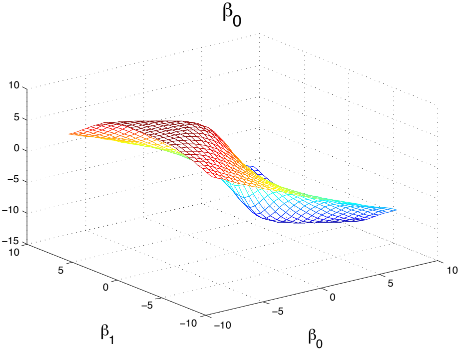
$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T}\right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$$

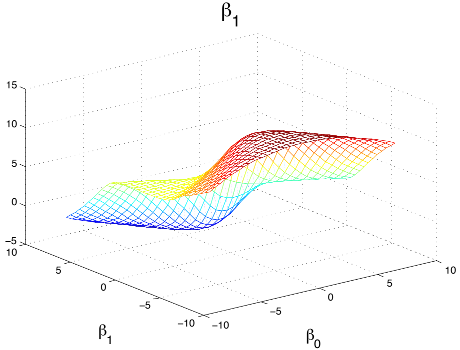
with derivatives evaluated at β^{old}

Logistic Regression

Let's look at

$$\frac{\partial \ell(\beta)}{\partial \beta} = \left[\frac{\partial \ell(\beta)}{\partial \beta_0}, \frac{\partial \ell(\beta)}{\partial \beta_1}\right]^T$$





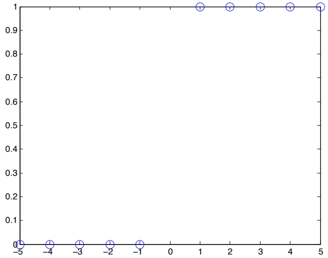
Newton-Raphson: $x^{new} = x^{old} - f''(x^{old})^{-1} f'(x^{old})$. Some of those slopes for f' look close to 0...

Logistic Regression

What could go wrong?

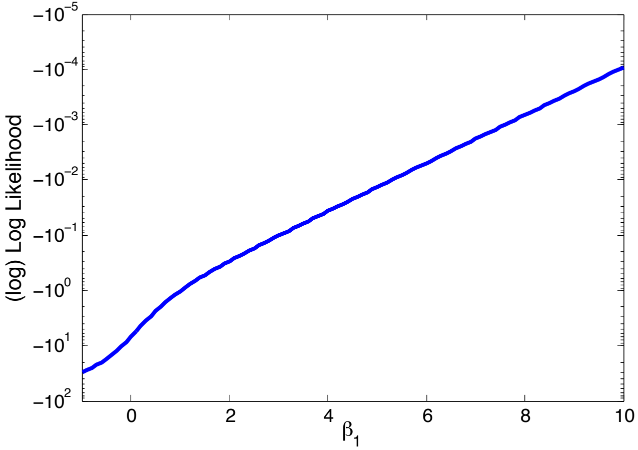
- log likelihood is concave, so we cannot get stuck in local optimum
- matrix inversion can get problematic when some eigenvalues are close to 0
- algorithm might not terminate (overshooting)
- anything else?

Consider data $x = (-5, -4, -3, -2, -1, 1, 2, 3, 4, 5)$ and $y = (0, 0, 0, 0, 0, 1, 1, 1, 1, 1)$



Logistic Regression

It can be shown that the ML estimate for β_0 is 0. The log likelihood for β_1 is...



Outline

Probit Regression

Problem: no easy solution to

$$\hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} y_i \log\left(\frac{e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}{1 + e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}\right) + (1 - y_i) \log\left(\frac{1}{1 + e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}\right)$$

We can also choose a link function where there is an easier solution!

Use a *probit* function:

$$\pi(x) = \Phi(x\beta) = \int_{-\infty}^{x\beta} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt$$

Note: $\Phi(x)$ is the cumulative density function for a Gaussian distribution, so if $X \sim N(0,1)$,

$$\Phi(x) = P(X \leq x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt$$

Probit Regression

Model:

$$
\begin{aligned}
\pi(x_i) \,|\, x_i &= \Phi(x\beta) \\
y_i \,|\, \pi(x_i) &\sim Bernoulli(\pi(x_i))
\end{aligned}
$$

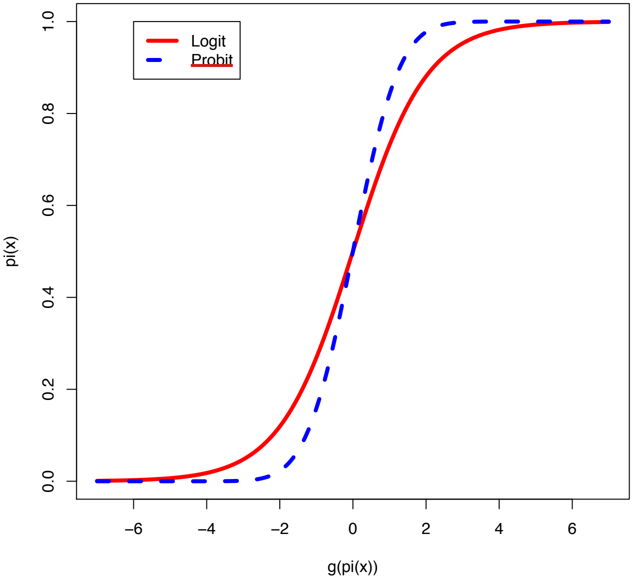
Model fitting: use maximum likelihood

$$
\ell(\beta) = \sum_{i=1}^{n} y_i \log\left(\Phi(x\beta)\right) + (1 - y_i) \log\left(1 - \Phi(x\beta)\right)
$$

$$
\hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} y_i \log\left(\Phi(x\beta)\right) + (1 - y_i) \log\left(1 - \Phi(x\beta)\right)
$$

Not a closed form solution, but finding optimal value is sometimes easier

Logit vs Probit Link Functions



Logistic vs Probit Regression

Logistic Regression	Probit Regression
$\pi(x) = \frac{e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}{1+e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}$	$\pi(x) = \Phi(x\beta)$
Heavier tails for $\pi(x)$	Lighter tails for $\pi(x)$
Harder inference for $\hat{\beta}$	Easier inference for $\hat{\beta}$

- logistic regression is more robust to outliers
- inference with probit regression is easier when there is collinearity, etc

Multinomial Logistic Regression

Say we have K categories instead of 2; use category K as the base variable

The new link function is defined by

$$P(Y_i = 1 \,|\, x) = \frac{e^{x\beta_{(1)}}}{1 + \sum_{k=1}^{K-1} e^{x\beta_{(k)}}}$$

$$\cdots\cdots$$

$$P(Y_i = K-1 \,|\, x) = \frac{e^{x\beta_{(K-1)}}}{1 + \sum_{k=1}^{K-1} e^{x\beta_{(k)}}}$$

$$P(Y_i = K \,|\, x) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{x\beta_{(k)}}}$$

Solution method for $\hat{\beta}$ is similar to that for logistic regression

Outline

Logistic Regression in R

Load South African Heart Disease data from *Elements of Statistical Learning* website

- predict whether patient has coronary heart disease from 9 predictors
- predictors: systolic blood pressure (sbp), cumulative tobacco use in kg (tobacco), LDL cholesterol (ldl), adiposity index (adiposity), family history of heart disease (present or absent, famhist), type-A behavior (typea), obesity given by BMI (obesity), current alcohol consumption (alcohol), age or age at onset (age)

```
> heart.df <- read.csv("heart.csv", header = TRUE)
> names(heart.df)
 [1] "sbp"       "tobacco"   "ldl"       "adiposity"
 [5] "famhist"   "typea"     "obesity"   "alcohol"
 [9] "age"       "chd"
```

Logistic Regression in R

To do logistic regression, we use the glm( ) function:

```
> attach(heart.df)
> fit.logit <- glm(chd ~ sbp + tobacco + ldl + adiposity + famhist
+ typea + obesity + alcohol + age, family="binomial")
> fit.logit$coefficients
  (Intercept)           sbp       tobacco
-6.1507208650  0.0065040171  0.0793764457
          ldl     adiposity famhistPresent
 0.1739238981  0.0185865682  0.9253704194
        typea       obesity       alcohol
 0.0395950250 -0.0629098693  0.0001216624
          age
 0.0452253496
```

Logistic Regression in R

We can also use the glm( ) function to do probit regression:

```
> fit.probit <- glm(chd ~ sbp + tobacco + ldl + adiposity
+ famhist + typea + obesity + alcohol + age,
family=binomial(link = "probit"))
> fit.probit$coefficients
   (Intercept)            sbp        tobacco
  -3.570182373    0.003789357    0.048219814
           ldl      adiposity famhistPresent
   0.102828873    0.012395619    0.538979697
         typea        obesity        alcohol
   0.023555723   -0.040162007    0.000019549
           age
   0.026269371
```