BT2101 Tutorial Week 11

Discrete Choice Model

Agenda

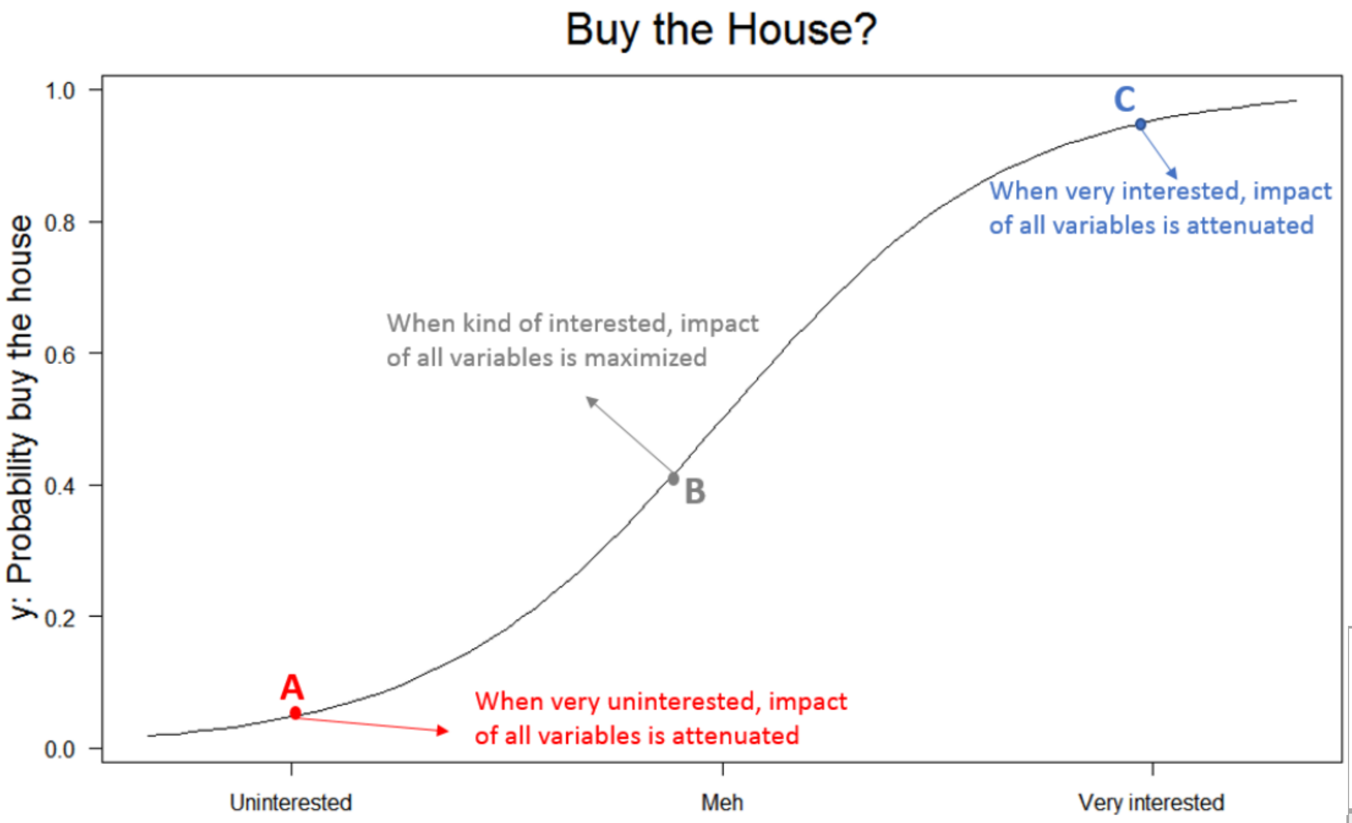
- Logit Model (Logistic Regression)

- Understand Logistic Regression
  - Logistic Distribution (i.e., Sigmoid Function)
  - Maximum Likelihood Estimation
  - Gradient Ascent/Descent

- Probit Model (Probability Unit)

- Homework 4 Solutions

Logistic Regression



$$\beta_0 + \beta_1 \times Size_i + \beta_2 \times Bedroom_i + \beta_3 \times Bathroom_i$$

$$\log\left(\frac{\Pr(Buy_i = 1)}{1 - \Pr(Buy_i = 1)}\right) = \beta_0 + \beta_1 \times Size_i + \beta_2 \times Bedroom_i + \beta_3 \times Bathroom_i$$

Example: Predict whether to buy a condo

Buy

(y=1, buy condo; y=0, otherwise)

3 features:

x={x_1: size, x_2: #bedroom, x_3: #bathroom}

Collect data from N=1000 people (i=1,…,1000):

Row	Size (m²)	#Bedroom	#Bathroom	Buy
1	65	2	2	1
2	72	3	2	0
……	……	……	……	……
1000	50	1	1	0

Logistic Regression

- Think about
  - General specification:

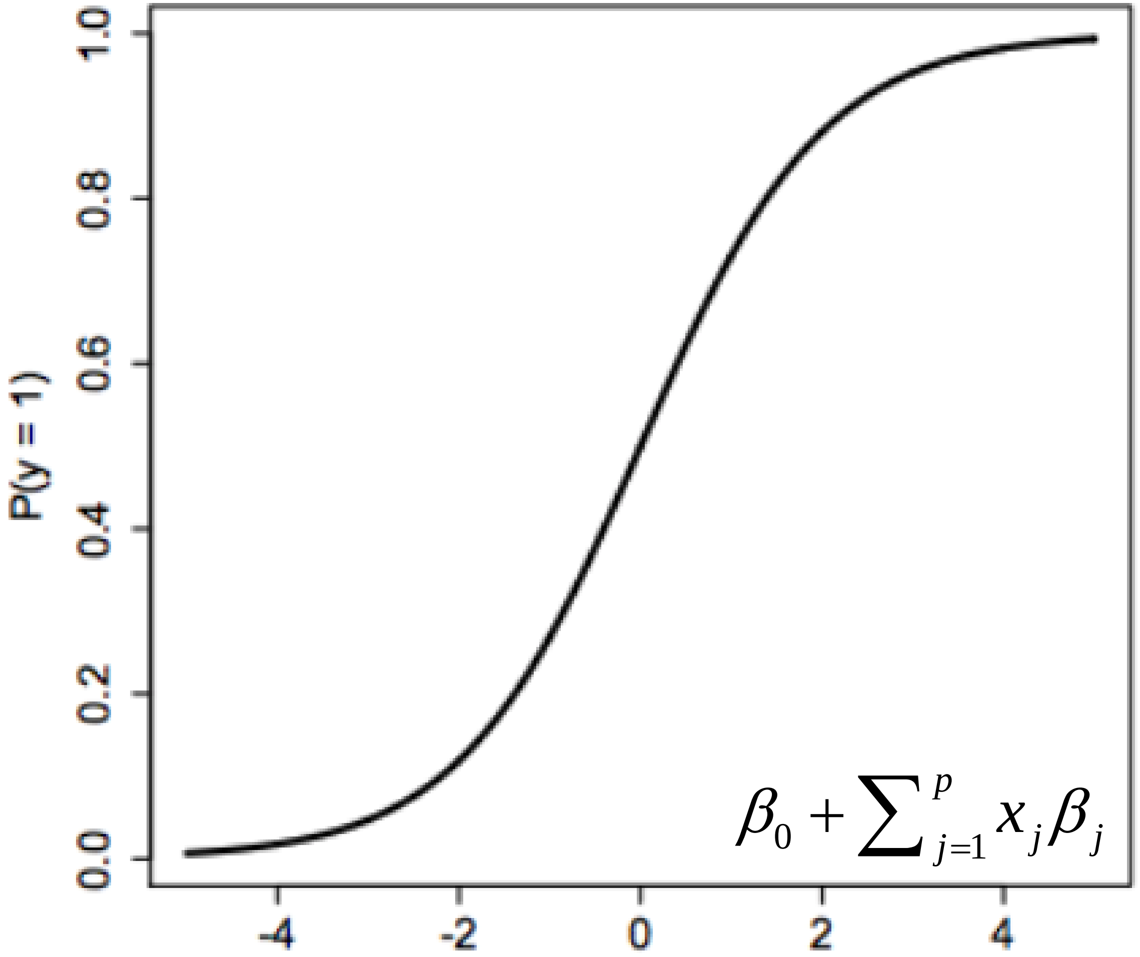
$$Logodds(\Pr(y_i = 1)) = \log(\frac{\Pr(y_i = 1)}{1 - \Pr(y_i = 1)})$$

$$= \beta_0 + \sum_{j=1}^{p} x_j \beta_j$$

$$\Pr(y_i = 1) = \frac{\exp(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)}$$

- Sigmoid function:

(Very Important Function)



Logistic Regression: Likelihood Function

We know score function: $\Pr(y_i = 1) = \dfrac{\exp(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)}$

and $\Pr(y_i = 0) = 1 - \Pr(y_i = 1)$

So the probability of the occurence of observation *i* (=1,...,N):

$$f(y_i) = [\Pr(y_i = 1)]^{(y_i)} [\Pr(y_i = 0)]^{(1-y_i)}$$

$$= [\frac{\exp(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)}]^{(y_i)} [\frac{1}{1 + \exp(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)}]^{(1-y_i)}$$

Logistic Regression: Likelihood Function

So the probability of the occurence of all the N observations:

Likelihood Function: $l(\beta) = \prod_{i=1}^{N} f(y_i)$

Score: ***Pr(y_i=1|x,β)***

$$= \prod_{i=1}^{N} \left[\boxed{\frac{e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}}{1 + e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}}}\right]^{y_i} \left[\boxed{\frac{1}{1 + e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}}}\right]^{(1-y_i)}$$

Pr(y_i=0|x,β)

Take log and we will get Log-Likelihood Function:

$$ll(\beta) = \sum_{i=1}^{N} [-\log(1 + e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}) + y_i(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)]$$

Estimation

- Gradient Ascent
  - Maximizing (Log-)Likelihood:

$$\max_{\beta} l(\beta) = \prod_{i=1}^{N} [\frac{e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}}{1 + e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}}]^{y_i} [\frac{1}{1 + e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}}]^{(1-y_i)}$$

$$\max_{\beta} ll(\beta) = \sum_{i=1}^{N} [-\log(1 + e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}) + y_i(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)]$$

  - Question: How to maximize this complex objective function?
    - Let first-order derivatives = 0 ? **Impossible.**
    - Maybe you can take steps by steps (i.e., **iteratively**) to approach the (local) optimal value

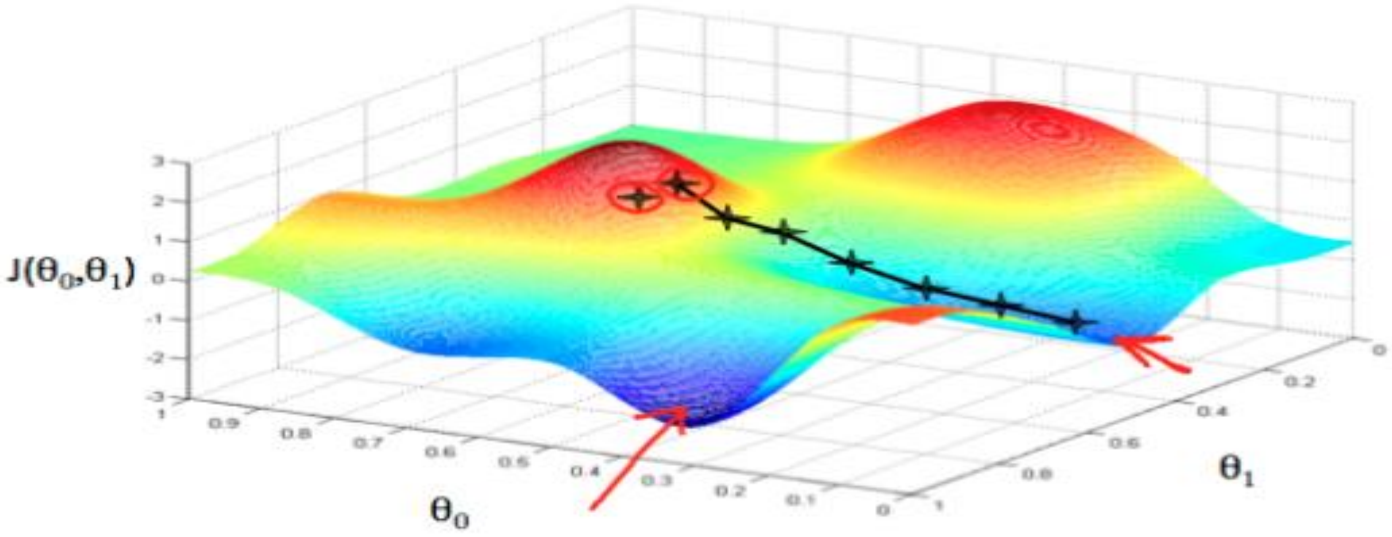
Remember What We've Learned in BT1101

Gradient Descent

(For Minimizing)

Gradient Ascent

(For Maximizing)



Correct: Simultaneous update

$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

$\theta_0 := \text{temp0}$

$\theta_1 := \text{temp1}$

Incorrect:

$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\theta_0 := \text{temp0}$

$\text{temp1} := \theta_1 - \boxed{\alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)}$

$\theta_1 := \text{temp1}$

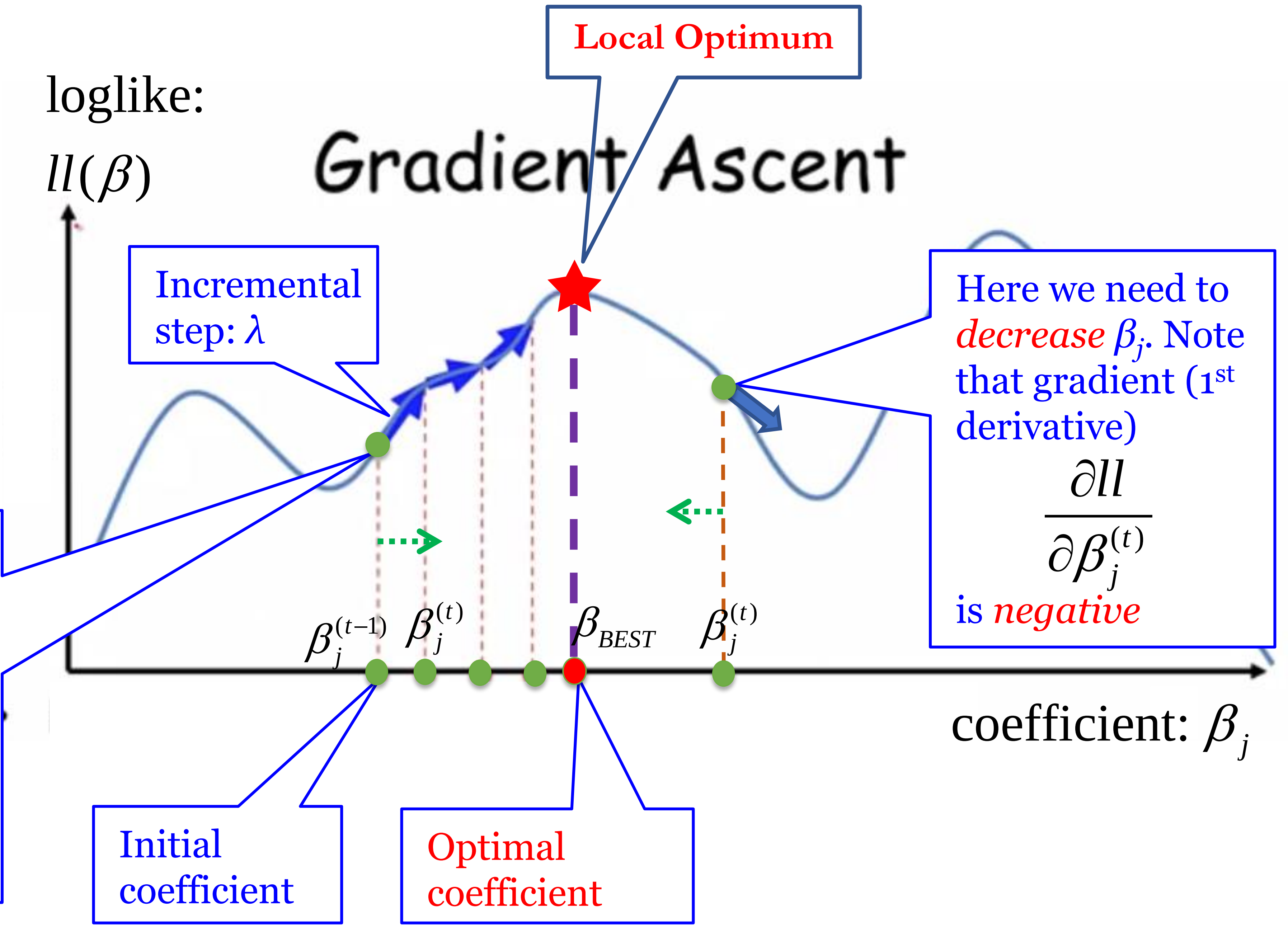
- gradient *descent* aims at *minimizing* some objective function: $\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$
- gradient *ascent* aims at *maximizing* some objective function: $\theta_j \leftarrow \theta_j + \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

Gradient Ascent

Update Rule:

Follow the direction as shown by gradient value; Move by a step with rate λ :

$$\beta_j^{(t)} \leftarrow \beta_j^{(t-1)} + \lambda \times \frac{\partial ll}{\partial \beta_j^{(t-1)}}$$



Here we need to *increase* β_j. Note that gradient (1st derivative)

$$\frac{\partial ll}{\partial \beta_j^{(t-1)}}$$

is *positive*

Update Rule

- Gradient Ascent
  - Maximizing (Log-)Likelihood:

$$ll(\beta) = \sum_{i=1}^{N} [-\log(1 + e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}) + y_i(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)]$$

Taylor Expansion

  - Gradient Ascent:
    - Remember **Taylor Expansion**
    - Newton-Raphson Method
      - Hard to get $-H_t^{-1}$
    - Steepest Ascent Method:
      - Let: $-H_t^{-1} = \lambda I$
      - λ: step size
      - I: Identity Matrix

8.3.1. Newton–Raphson

To determine the best value of β_{t+1}, take a second-order Taylor's approximation of $LL(\beta_{t+1})$ around $LL(\beta_t)$:

(8.1)

$$LL(\beta_{t+1}) = LL(\beta_t) + (\beta_{t+1} - \beta_t)' g_t + \tfrac{1}{2}(\beta_{t+1} - \beta_t)' H_t (\beta_{t+1} - \beta_t).$$

Now find the value of β_{t+1} that maximizes this approximation to $LL(\beta_{t+1})$:

$$\frac{\partial LL(\beta_{t+1})}{\partial \beta_{t+1}} = g_t + H_t(\beta_{t+1} - \beta_t) = 0,$$

$$H_t(\beta_{t+1} - \beta_t) = -g_t,$$

$$\beta_{t+1} - \beta_t = -H_t^{-1} g_t,$$

$$\beta_{t+1} = \beta_t + \boxed{(-H_t^{-1})} g_t.$$

Gradient Ascent

$$\beta_{t+1} \leftarrow \beta_t + \lambda \times g_t$$

Gradient Ascent

- Gradient Ascent
  - Objective Function: $ll(\beta) = \sum_{i=1}^{N} [-\log(1+e^{\beta_0+\sum_{j=1}^{p} x_j\beta_j}) + y_i(\beta_0 + \sum_{j=1}^{p} x_j\beta_j)]$
  - Gradient Ascent:
    - (1) Initialize $\beta^{(0)} = (\beta_0^{(0)}, \beta_1^{(0)}, ..., \beta_j^{(0)}) = (0, 0, ..., 0), t = 1$
    - (2) In step ***t***, update coefficients:

$$\frac{\partial ll}{\partial \beta_j} \leftarrow \sum_{i=1}^{N} (y_i - \frac{e^{\beta_0^{(t-1)}+\sum_{j=1}^{p} x_j\beta_j^{(t-1)}}}{1+e^{\beta_0^{(t-1)}+\sum_{j=1}^{p} x_j\beta_j^{(t-1)}}}) x_{ij}$$ Calculate gradients (first-order derivatives) of coefficients

$$\beta_j^{(t)} \leftarrow \beta_j^{(t-1)} + stepsize \times \frac{\partial ll}{\partial \beta_j}$$ Update coefficients with Steepest Ascent Method

$$t \leftarrow t+1$$

    - (3) Check convergence condition $\|\nabla ll(\beta^{(t)})\| < tolerance$. If not, repeat (2) until (3) is satisfied

Gradient Ascent

- Gradient Ascent:
  - Python codes: https://github.com/mozartkun/IS4303_Tutorials_2019_SEM2/blob/master/Tutorial%203.%20Data%20Preprocessing%20and%20Linear%20Model/IS4303%20Tutorial%20Week4%20Simplified%20Version.ipynb

Logistic Regression: Interpretation

Given a logistic model: $Logodds(\Pr(y_i = 1)) = \log(\frac{\Pr(y_i = 1)}{1 - \Pr(y_i = 1)}) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j$

How to interpret coefficient β_j ?

(1) Marginal effect of x_j on log(odds_ratio): $\log(\frac{\Pr(y_i = 1)}{1 - \Pr(y_i = 1)})$

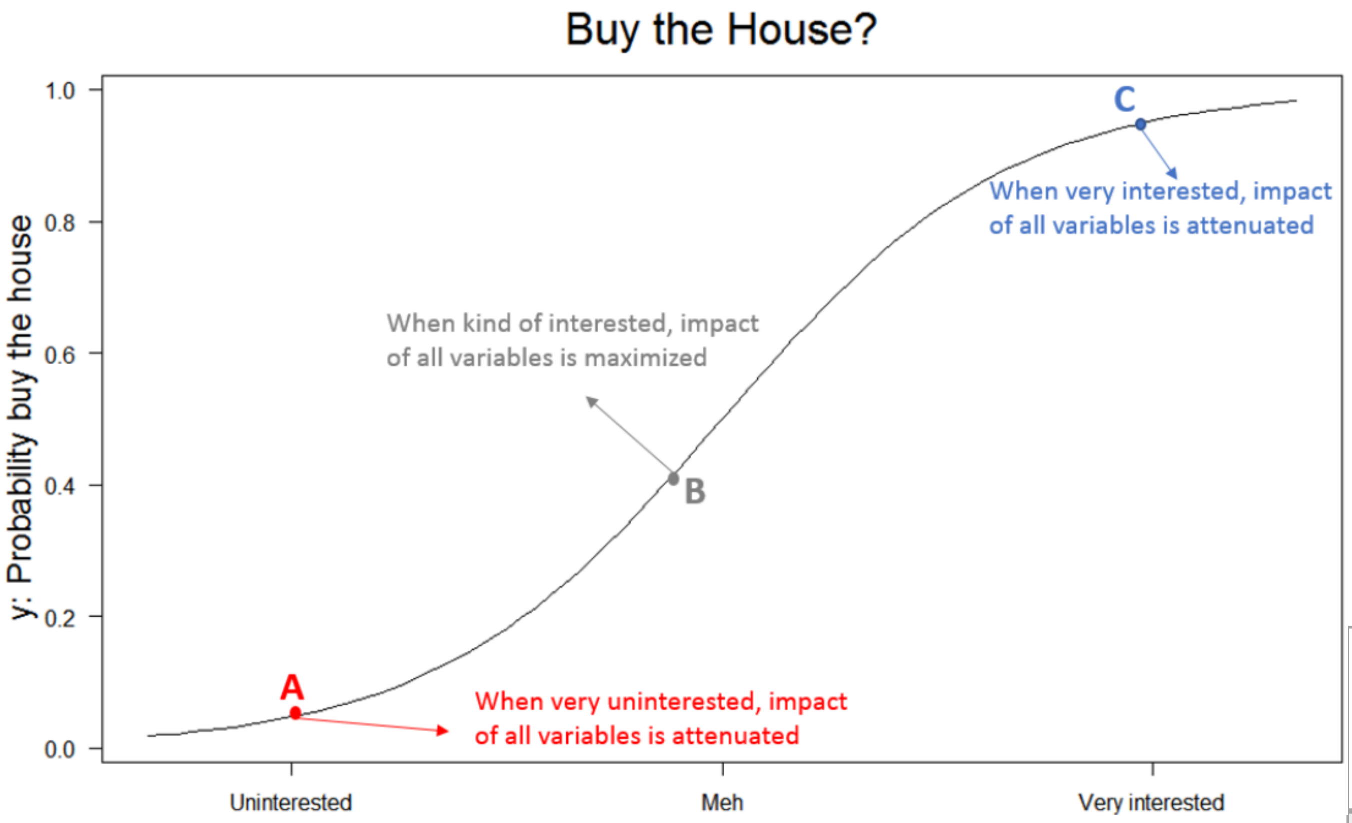
(2) $\frac{\Pr(y_i = 1)}{1 - \Pr(y_i = 1)} = e^{\beta_0 + \sum_{j=1}^{p} x_j \beta_j}$:

x_j increases by 1 unit, original odds_ratio is multiplied by a factor e^{β_j}

(3)Doubling Amout:

x_j increases by $\frac{\ln(2)}{\beta_j}$ unit, original odds_ratio is doubled

Probit Regression



$\beta_0 + \beta_1 \times Size_i + \beta_2 \times Bedroom_i + \beta_3 \times Bathroom_i$

$$\Pr(Buy_i = 1) = \Phi(\beta_0 + \beta_1 \times Size_i + \beta_2 \times Bedroom_i + \beta_3 \times Bathroom_i)$$

Example: Predict whether to buy a condo

Buy

(y=1, buy condo; y=0, otherwise)

3 features:

x={x_1: size, x_2: #bedroom, x_3: #bathroom}

Collect data from N=1000 people (i=1,…,1000):

Row	Size (m²)	#Bedroom	#Bathroom	Buy
1	65	2	2	1
2	72	3	2	0
……	……	……	……	……
1000	50	1	1	0

Probit Regression

- Assume there is a latent unobservable utility y^* (e.g., Level of happiness):

$$y_i^* = \beta_0 + \sum_{j=1}^{p} x_j\beta_j + \varepsilon \quad \text{where the outcome: } y_i = \begin{cases} 1, & \text{if } y_i^* \geq 0 \\ 0, & \text{if } y_i^* < 0 \end{cases}$$

$$\Pr(y_i = 1) = \Pr(y_i^* \geq 0) = \Pr(\beta_0 + \sum_{j=1}^{p} x_j\beta_j + \varepsilon \geq 0) = \Pr(\varepsilon \geq -c) \text{ where } c = \beta_0 + \sum_{j=1}^{p} x_j\beta_j$$

- Parametric Assumption on random error ε
  - Logistic Distribution CDF (i.e., Sigmoid Function):

$$\Pr(\varepsilon \geq -c) = \frac{e^c}{1+e^c} \Rightarrow \Pr(y_i = 1) = \frac{e^{\beta_0 + \sum_{j=1}^{p} x_j\beta_j}}{1+e^{\beta_0 + \sum_{j=1}^{p} x_j\beta_j}} \Rightarrow \text{Logistic Regression}$$

  - Standard Normal Distribution CDF $\Phi(\cdot)$:

$$\Pr(\varepsilon \geq -c) = \Pr(\varepsilon \leq c) \Rightarrow \Pr(y_i = 1) = \Phi(\beta_0 + \sum_{j=1}^{p} x_j\beta_j) \Rightarrow \text{Probit Regression}$$

Estimation

- Gradient Ascent
  - Maximizing (Log-)Likelihood:

$$\max_{\beta} l(\beta) = \prod_{i=1}^{N} \Phi(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)^{y_i} [1 - \Phi(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)]^{(1-y_i)}$$

$$\max_{\beta} ll(\beta) = \sum_{i=1}^{N} [y_i \ln \Phi(\beta_0 + \sum_{j=1}^{p} x_j \beta_j) + (1 - y_i) \ln(1 - \Phi(\beta_0 + \sum_{j=1}^{p} x_j \beta_j))]$$

  - Question:
    - Computationally expensive in calculating $\Phi(\beta_0 + \sum_{j=1}^{p} x_j \beta_j)$
    - Numerical Approximation: https://en.wikipedia.org/wiki/Normal_distribution

Homework 4 Solutions

Thank you!