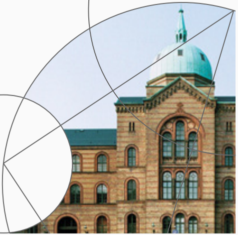
Binary Response: The Probit and Logit Models

Advanced Microeconometrics

Anders Munk-Nielsen

2022



Plan for lectures: Helicopter

Part I: Linear methods. ✓

Part II: High-dimensional methods. ✓

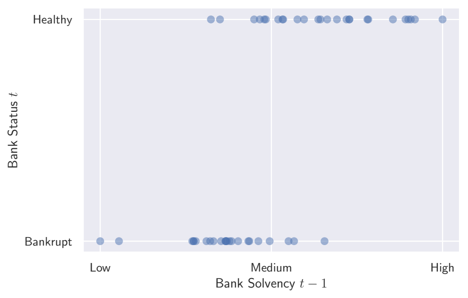
Part III: M-estimation, theory ✓

Part IV: M-estimation, concrete models ←

Where are we in the course?

Part	Topic	Parameterization non-linear	Estimation non-linear	Dimension dim(x)	Numerical optimization	M-estimation (Part III)	Outcome (y_i)	Panel (c_i)
I	OLS	÷	÷	low	÷	✓	$\mathbb{R}$	✓
II	LASSO	÷	✓	high	✓	÷	$\mathbb{R}$	÷
IV	Probit	✓	✓	low	✓	✓	$\{0,1\}$	÷
	Tobit	✓	✓	low	✓	✓	$[0;\infty)$	÷
	Logit	✓	✓	low	✓	✓	$\{1,2,\ldots,J\}$	÷
	~~Sample selection~~	~~✓~~	~~✓~~	~~low~~	~~✓~~	~~✓~~	~~$\mathbb{R}$ and $\{0,1\}$~~	~~÷~~
	Simulated Likelihood	✓	✓	low	✓	✓	Any	✓
	Quantile Regression	÷	✓	(low)	✓	✓	$\mathbb{R}$	÷
	Non-parametric	✓	(✓)	∞	÷	÷	$\mathbb{R}$	÷

Blitz discussion



Discuss

Suppose the data above shows *historical* bank solvency against whether the bank survives or goes bankrupt. And suppose the social planner wants to maximize expected gain from tax payer money.

- **Q:** Going forward, which banks should we bail out?

Today's goal

Goals: Code-wise

For all M-estimators, we will write `model.py` with the key ingredients

```
def q(theta, y, x):
        # FILL IN
        return -loglike # N-vector

def starting_values(y, x):
        # FILL IN
        return theta0 # K-vector

def sim_data(theta, N):
        # FILL IN
        return y, x

# to estimate a model (probit, logit, tobit, etc.)
theta0 = model.starting_values(y, x)
result = estimation.estimate(model.q, theta0, y, x)
```

(Common for: `probit.py`, `logit.py`, `clogit.py`, `tobit.py`, `qreg.py`)

Non-linear Models

- **Outcome and associated models**

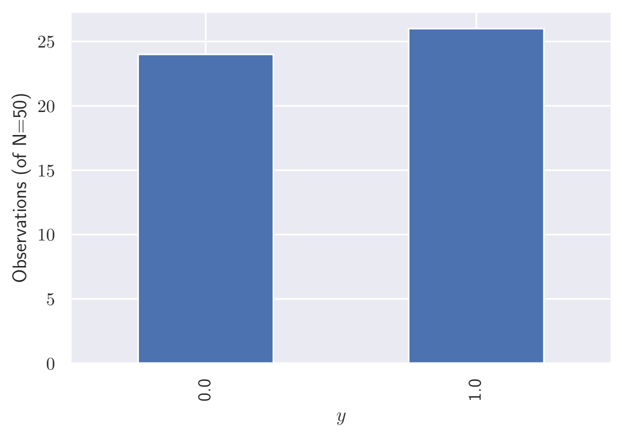
Outcome	**Name**	**Model**
$y \in \{0, 1\}$	Binary	Probit, Logit
$y \in \{0, 1, ..., J\}$	Unordered	Conditional/multinomial logit
$y \in [0; \infty)$	Censored	Tobit
$y \in \mathbb{N}$	Count data	[not covered]

- **Methodology:**
  - Write up a model (the DGP)
  - Derive the likelihood function
  - Estimate parameters
- **Today:** Binary response.

Agenda

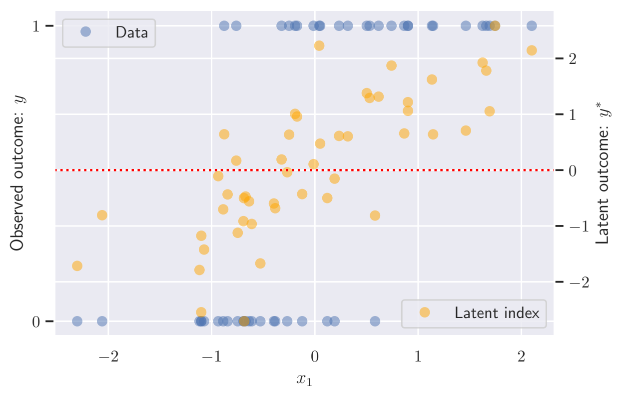
Outline

Data



Data: y vs. $x\beta$

Data: y vs. $x\beta$



Latent Variable Model

$$\begin{aligned} y_i^* &= \mathbf{x}_i \boldsymbol{\beta}_o + \varepsilon_i, \quad \varepsilon_i \sim G_o(\cdot), \\ y_i &= \mathbf{1}\{y_i^* > 0\}. \end{aligned}$$

- **where**
  - y_i^* is the *latent* unobserved index,
  - we either observe $y_i = 1$ or $y_i = 0$,
  - $G_o(\cdot)$ is the (true) cdf of ε_i, i.e. $\Pr(\varepsilon_i \leq z) = G_o(z)$,
  - $\mathbf{1}\{\cdot\}$ is an *indicator*, (=1 if the event is true, =0 otherwise).

Latent Variable Model

Latent Variable Model

$$
\begin{aligned}
y_i^* &= \mathbf{x}_i \beta_o + \varepsilon_i, \quad \varepsilon_i | \mathbf{x}_i \sim G_o(\cdot), \\
y_i &= \mathbf{1}\{y_i^* > 0\}.
\end{aligned}
$$

- **Task:** show $\Pr(y_i = 1 | x_i) = G_o(\mathbf{x}_i' \beta_o)$ if G_o is symmetric.

Intermezzo: Drawing from $G(\cdot)$

Drawing from a density

If $U \sim \mathrm{Uniform}(0,1)$, then the variable

$$V := G^{-1}(U),$$

will have cdf $G(\cdot)$,
where $G^{-1}(\cdot)$ is the inverse of $G(\cdot)$.

```
import numpy as np
from scipy.stats import norm, logistic
U = np.random.uniform(size=1000)
X = norm.ppf(U) # X is standard normal
Z = logistic.ppf(U) # Z is standard logistic
```

Latent Variable Model

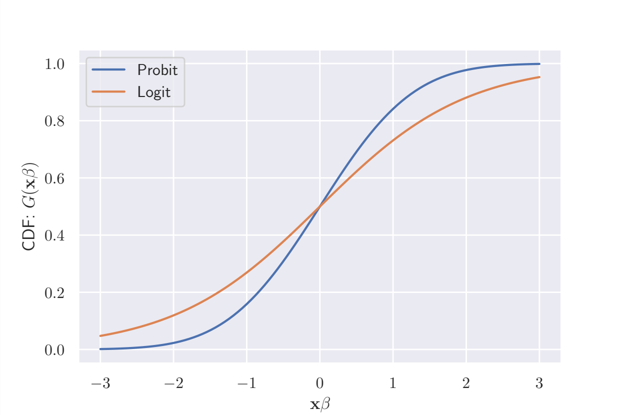
$$
\begin{aligned}
y_i^* &= \mathbf{x}_i\beta_o + \varepsilon_i, \quad \varepsilon_i|\mathbf{x}_i \sim G_o(\cdot), \\
y_i &= \mathbf{1}\{y_i^* > 0\}.
\end{aligned}
$$

```
function sim_data(N,theta):
    # 1. simulate x variables.
    oo = np.ones((N,1))
    xx = np.random.normal(size=(N,K-1))
    x = np.hstack([oo, xx]);
    # 2 draw error terms
    uniforms = np.random.uniform(size=N)
    u = Ginv(uniforms)
    # 2 compute latent index
    ystar = x@beta + u
    # 2 compute observed y (as a float)
    y = (ystar>=0).astype(float)
    return y,x # ystar is not observed
```

Common choices of $G(\cdot)$

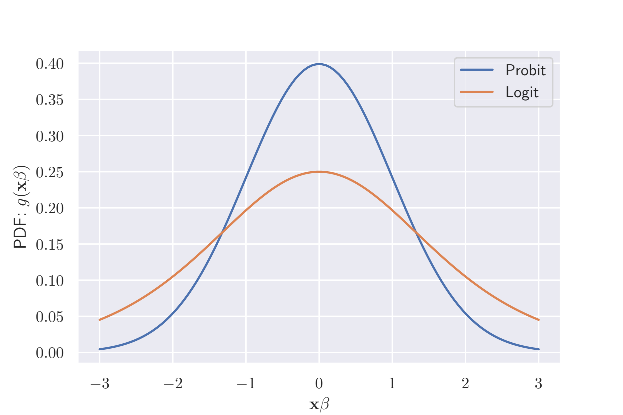
- **Probit:** $G(z) = \Phi(z)$ [standard normal cdf]
- **Logit:** $G(z) = \frac{1}{1+\exp(-z)}$ [logistic cdf]

Logit vs. Probit




```python
from scipy.stats import norm, logistic
G = lambda z : norm.cdf(z) # probit
G = lambda z : logistic.cdf(z) # logit
G = lambda z : 1.0 / (1.0 + np.exp(-z)); # logit (analytic)
```

Logit vs. Probit




```python
from scipy.stats import norm, logistic
G = lambda z : norm.pdf(z) # probit
G = lambda z : logistic.pdf(z) # logit
```

Outline

Deriving the Likelihood

- **Note** that y is Bernoulli with $\Pr(y = 1|\mathbf{x}) = G(\mathbf{x}\beta_o)$.
- **Likelihood:**

$$\log f(y|\mathbf{x}) = \mathbf{1}_{\{y=1\}} \log \Pr(y = 1|\mathbf{x}) + \mathbf{1}_{\{y=0\}} \log \Pr(y = 0|\mathbf{x}).$$

  - and $\Pr(y = 0|\mathbf{x}) = 1 - \Pr(y = 1|\mathbf{x})$.
- **Criterion:**

$$q(y_i, \mathbf{x}_i, \beta) = -\mathbf{1}_{\{y=1\}} \log G(\mathbf{x}\beta) - \mathbf{1}_{\{y=0\}} \log[1 - G(\mathbf{x}\beta)].$$

Discuss

When do we get consistency when we minimize $\sum_i q(\cdot)$?

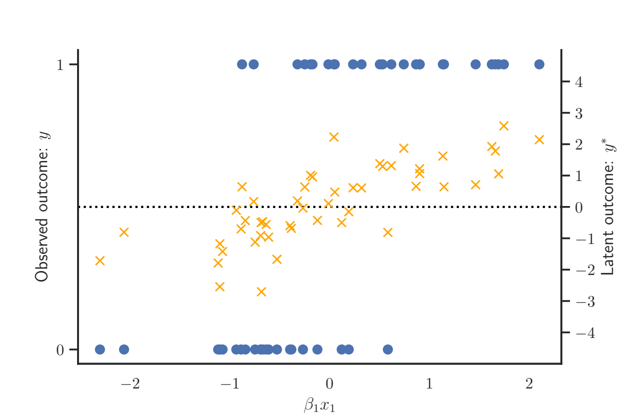
Identification

Definition: Identification

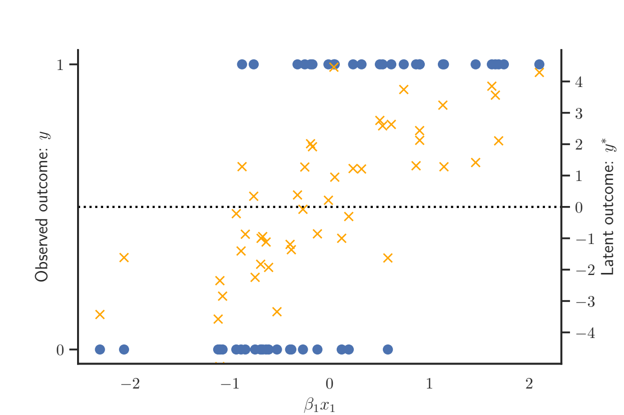
We say that the parameters of the model are *identified* if there exists a unique θ_o that minimizes the population criterion, $Q_o(\theta) \equiv \mathbb{E}[q(w, \theta)]$.

Appropriate definition for M-estimators.

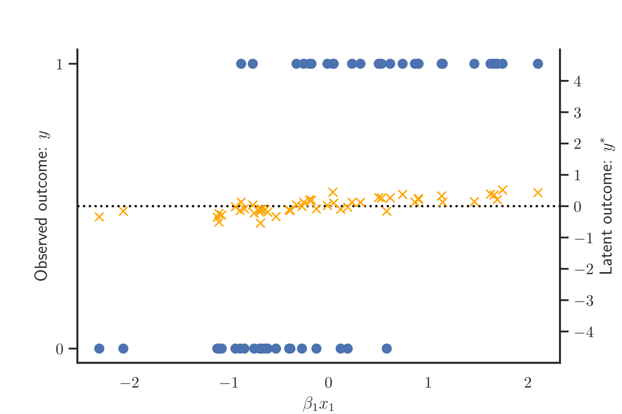
Identification intuition: Default (β, σ)



Identification intuition: $(2\beta, 2\sigma)$



Identification intuition: $\left(\frac{1}{5}\beta, \frac{1}{5}\sigma\right)$



A "Gaussian" Model, 1

scale of the error term

$$\begin{aligned} y_i^* &= \mathbf{x}_i \boldsymbol{\beta}_o + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_o^2). \\ y_i &= \mathbf{1}\{y_i^* > 0\}. \end{aligned}$$

This yields the criterion

$$q(y_i, \mathbf{x}_i, \beta, \sigma) = -\mathbf{1}_{\{y=1\}} \log \Phi\left(\frac{\mathbf{x}\beta}{\sigma}\right) - \mathbf{1}_{\{y=0\}} \log\left[1 - \Phi\left(\frac{\mathbf{x}\beta}{\sigma}\right)\right].$$

Discuss (scale normalization)

Which of these sets of parameters shows non-identification?

1. $\boldsymbol{\theta} = (\boldsymbol{\beta}_o + k, \sigma_o + k)'$.
2. $\boldsymbol{\theta} = (k\boldsymbol{\beta}_o, k\sigma_o)'$, <- This guy. This shows there is NOT a unique solution to the criterion func see the def of identification
3. $\boldsymbol{\theta} = (k\boldsymbol{\beta}_o, \frac{1}{k}\sigma_o)'$,

Example: Gaussian Model

A “Gaussian” Model, 2

$$
\begin{aligned}
y_i^* &= \mathbf{x}_i \beta_o + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(\mu_0, 1). \\
y_i &= \mathbf{1}\{y_i^* > 0\}.
\end{aligned}
$$

This yields the criterion

$$
q(y_i, \mathbf{x}_i, \beta, \mu) = -\mathbf{1}_{\{y=1\}} \log \Phi\left(\mathbf{x}\beta - \mu\right) - \mathbf{1}_{\{y=0\}} \log\left[1 - \Phi\left(\mathbf{x}\beta - \mu\right)\right].
$$

Discuss (location normalization)

Why is μ not identified?

`if you add \mu to the constant in x - they must sum to the same to show that \mu is not identified`

`if we have a constant in x, then`

The Probit Model

$$y_i^* = \mathbf{x}_i \beta_o + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0,1).$$
$$y_i = \mathbf{1}\{y_i^* > 0\}.$$

The Probit Criterion

$$q(y_i, \mathbf{x}_i, \beta) = -\mathbf{1}_{\{y=1\}} \log \Phi(\mathbf{x}\beta) - \mathbf{1}_{\{y=0\}} \log[1 - \Phi(\mathbf{x}\beta)].$$

```
from scipy.stats import norm
def q(theta, y, x):
        xb = x @ theta
        Gxb = norm.cdf(xb)
        return -(y == 1) * np.log(Gxb) - (y == 0) * np.log(1.0 - Gx
```

Outline

Conditional Distribution

- **Model:** For $G(\cdot,\cdot)$ known, assume

$$\Pr(y_i = 1|\mathbf{x}_i) = G(\mathbf{x}_i, \boldsymbol{\beta}_o).$$

- **Question:** What is $\mathbb{E}(y|x_i)$?
  - **Answer:** $\mathbb{E}(y|\mathbf{x}_i) =$ probability of success ("G-index")
- **Question:** What is $\mathrm{Var}(y|\mathbf{x}_i)$?
  - **Answer:** $\mathrm{Var}(y|x_i) =$ product of success and failure
- **Question:** Which estimators could be used?
  - **Answer:** OLS, MLE (Probit/Logit)

Which model to use?

	Probit	Logit	"OLS" LPM
β_1	0.256	0.462	0.529
β_2	1.656	2.853	0.348

Challenge: How do we compare across models?

Answer: Compare *partial effects*, not the underlying parameters.

Partial Effects

- **Model:**

$$\Pr(y = 1|\mathbf{x}) = \mathbb{E}(y|\mathbf{x}) = G(\mathbf{x}_i\boldsymbol{\beta}_o), \quad G'(\cdot) \equiv g(\cdot).$$

- **Note:** Magnitude of β_k is hard to interpret.
  - **Intuition:** Measured in "utils".
- **Object of interest?** Partial effects of x_{ip} on the *response probability*.
  - **OLS:** $\partial\mathbb{E}(y|\mathbf{x})/\partial x_p = \beta_p$
  - ... **doesn't depend on** x**!**
- **Generally:** (and here) the PEs must be evaluated at some $\mathbf{x} = \mathbf{x}^0 \equiv (x_1^0, ..., x_P^0)$.
- **Dummy** x_p**?** Then the derivative doesn't make sense intuitively...
  - **Solution:** Differences.

Partial Effects

- **Continuous x_p:** The PE of x_p at $\mathbf{x}^0$,

$$\delta_p(x^0) \equiv \left.\frac{\partial \Pr(y = 1|\mathbf{x})}{\partial x_p}\right|_{\mathbf{x}=\mathbf{x}^0} \quad = \quad g(\mathbf{x}^0\boldsymbol{\beta})\beta_p.$$

  - **Probit:** $\delta_p(\mathbf{x}^0) = \phi(\mathbf{x}^0\boldsymbol{\beta})\beta_p.$

    lower case \phi -> normal pdf

- **Binary x_p (dummy):** Let $x^0(x_p = i)$ denote $\mathbf{x^0}$ with x_p set to the value i. Then

$$\begin{aligned} \delta_p(x^0) \quad &\equiv \quad \Pr\left[y = 1|\mathbf{x}^0(x_p = 1)\right] - \Pr\left[y = 1|\mathbf{x}^0(x_p = 0)\right] \\ &= \quad G\left[\mathbf{x}^0(x_p = 1)\boldsymbol{\beta}\right] - G\left[\mathbf{x}^0(x_p = 0)\boldsymbol{\beta}\right]. \end{aligned}$$

Partial Effects

- **Partial effect:** Let $g = G'$. The partial effect, δ_p, is

small partial effects at the boundary
close to zero in a pdf, the partial effect is largest

$$\delta_p(\mathbf{x}^0) = \begin{cases} g(\mathbf{x}^0\beta)\beta_p & \text{if } x_p \text{ is continuous,} \\ G\left(\mathbf{x}^1\beta\right) - G\left(\mathbf{x}^0\beta\right) & \text{if } x_p \text{ is a dummy,} \end{cases}$$

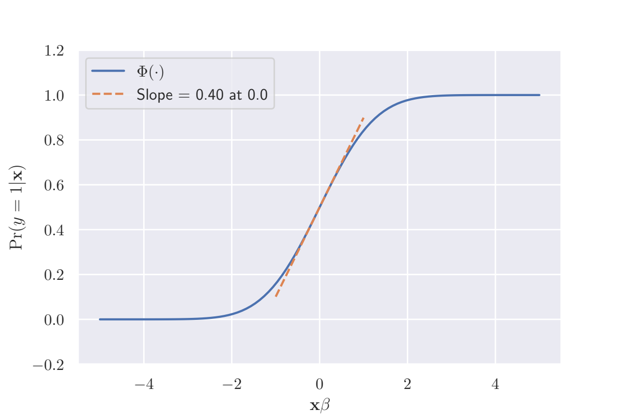
where $x_p^1 = 1$ and $x_p^0 = 0$ if x_p is a dummy (and $x_k^1 = x_k^0$ for $k \neq p$).

1. **Sign:** β_p determines whether $\delta_p \gtrless 0$.
2. **Depends on $\mathbf{x}^0$:** Typically, $\mathbf{x}^0$ is the average or median characteristics.
   - **Average PE:** Alternatively, average across observations:

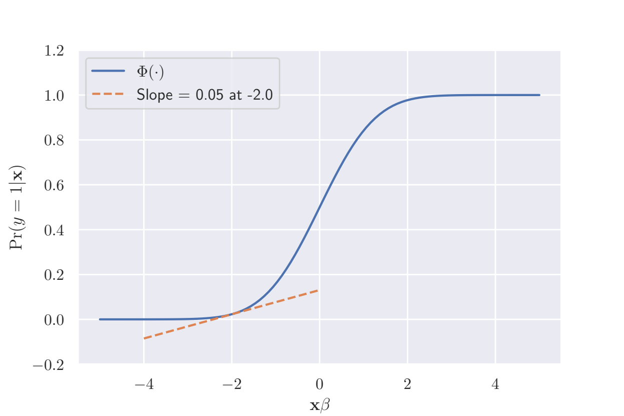
$$\text{APE}_p = N^{-1}\sum_{i=1}^{N} \delta_p(\mathbf{x}_i).$$

3. **Largest** when $\mathbf{x}^0\beta \cong 0$; smaller when $|\mathbf{x}^0\beta|$ is large.
   - **Mathematically:** g is a pdf; $g(z) \to 0$ as $z \to \pm\infty$.
   - **Example:** Job training programs' effectiveness depends on baseline probability.

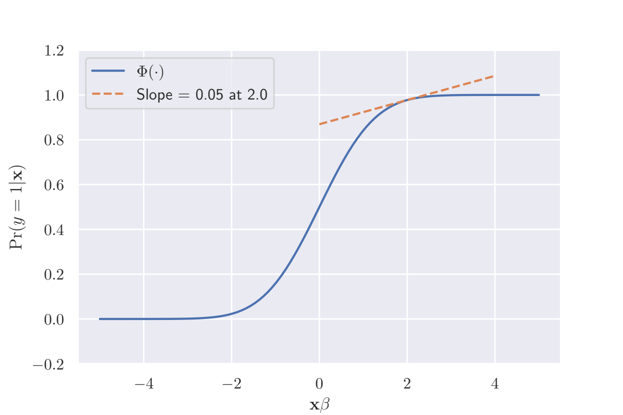
Partial Effects



Partial Effects



Partial Effects



Partial effects (simulation example)

	Probit	Logit	LPM	
Marginal effect of x_1	0.054	0.056	0.529	discrete variable
Marginal effect of x_2	0.351	0.348	0.348	cont. variable

- **Intuition:** The partial effect is really what we are after.

Remark: Identification of Partial Effects

- Suppose $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, $\sigma \neq 1$.
- Then

$$\Pr(\varepsilon > -\mathbf{x}\beta|\mathbf{x}) = \Pr(\varepsilon/\sigma > -\mathbf{x}\beta/\sigma|\mathbf{x}) = \Phi(\mathbf{x}\beta/\sigma).$$

- So the partial effect becomes

$$\delta_p(x^0) = \phi\left(\frac{\mathbf{x}^0\boldsymbol{\beta}}{\sigma}\right)\frac{\beta_p}{\sigma}, \quad p = 1, ..., \underbrace{P}_{\equiv \dim \boldsymbol{\beta}}$$

Conclusion: Only the "normalized" coefficients, β_p/σ, matter for the partial effects.

- $\Rightarrow$ The normalization $\sigma := 1$ is without loss of generality [if the interest is in partial effects]

- **Problem:** We can estimate $\mathrm{Var}(\hat{\beta}_p)$, but what is $\mathrm{Var}(\hat{\delta}_p)$?
- **Note:** $\hat{\delta}_p$'s are a function of $\hat{\beta}$. Suppressing dependence on x^0,

$$\hat{\delta}_p = g(\mathbf{x}^0\hat{\beta})\hat{\beta}_p \equiv h(\hat{\beta})$$

the asymptotic variance of h is almost

Delta Method

$$\mathrm{Avar}[h(\hat{\beta})] \overset{\text{approx}}{\cong} \left[\nabla h(\hat{\beta})\right] \mathrm{Avar}(\hat{\beta}) \left[\nabla h(\hat{\beta})\right]'.$$

- **Intuition:** The variance of $\hat{\beta}_q$ affects the variance of $\hat{\delta}_p$ more if $\hat{\beta}_q$ is important in h (i.e. $h_q'(\hat{\beta})$ is large).
- **Hands-on:** Covered in one of the exercise classes.

Outline

How to choose?

- **Typically:** Not much difference.
  - ... same partial effects are identified.
  - But parameter values differ.
- **Estimation:** Possible to estimate $G(\cdot)$ non-parametrically,
  - e.g. Klein & Spady (1993), Manski's maximum score (LAD).
  - Don't always work that well in practice;
  - ... appears to not make a huge difference *at the mean*.

Probit vs. Logit

Probit

$$y_i^* = \mathbf{x}_i\boldsymbol{\beta}_o + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0,1)$$

$$\Pr(y_i = 1|\mathbf{x}_i) = \Phi(\mathbf{x}_i\boldsymbol{\beta}_o)$$

`partial effect`
$$\frac{\partial \Pr(y=1|x)}{\partial x_p}(\mathbf{x}^0) = \phi(x^0\beta)\beta_p.$$

Logit

$$y_i^* = \mathbf{x}_i\boldsymbol{\beta}_o + \varepsilon_i, \quad \varepsilon_i \sim \text{Logistic}$$

$$\Pr(y_i = 1|\mathbf{x}_i) = \frac{1}{1+\exp(-\mathbf{x}_i\boldsymbol{\beta}_o)}$$

`partial effect`
$$\frac{\partial \Pr(y=1|\mathbf{x})}{\partial x_p}(\mathbf{x}^0) = \frac{\exp(-\mathbf{x}_i\boldsymbol{\beta}_o)}{[1+\exp(-\mathbf{x}_i\boldsymbol{\beta}_o)]^2}\beta_p.$$

Probit vs. Logit

- **Distributions:** Logistic has slightly fatter tails.
- **Relative PEs:** Recall,

$$\delta_p = g(\mathbf{x}^0\boldsymbol{\beta})\beta_p.$$

- **If** $\mathbf{x}^{0\prime}\boldsymbol{\beta} \cong 0$ for both models,

$$\begin{aligned} \delta_p^{\text{Logit}} &= g^{\text{Logit}}(0)\beta_p^{\text{Logit}} = \frac{1}{4}\beta_p^{\text{Logit}} \\ \delta_p^{\text{Probit}} &= \phi(0)\beta_p^{\text{Probit}} = \frac{1}{\sqrt{2\pi}}\beta_p^{\text{Probit}}. \end{aligned}$$

- **PEs identified:** Hence $\delta_p^{\text{Logit}} \cong \delta_p^{\text{Probit}}$

$$\begin{aligned} \Rightarrow \frac{1}{4}\beta_p^{\text{Logit}} &\cong \frac{1}{\sqrt{2\pi}}\beta_p^{\text{Probit}} \\ \Leftrightarrow \beta_p^{\text{Logit}} &\cong 1.6\beta_p^{\text{Probit}}. \end{aligned}$$

- **Exercise class:** Verify this.

Semi-parametric Specification (not curriculum)

- **Possible** to avoid functional assumptions, leaving G free.
- **We derived** the likelihood function:

$$\ell_i(\boldsymbol{\theta}) = y_i \log G(\mathbf{x}_i\beta) + (1 - y_i) \log [1 - G(\mathbf{x}_i\beta)] \, .$$

Klein & Spady Criterion

$$\ell_i(\boldsymbol{\theta}) = y_i \log \hat{G}(\mathbf{x}_i\beta) + (1 - y_i) \log \left[1 - \hat{G}(\mathbf{x}_i\beta)\right] ,$$

where $\hat{G}(\cdot)$ is computed using Kernel methods (lecture #17).

- **Normalizations:** In Probit, we assume $\mu = 0, \sigma = 1$; here, $\hat{G}$ is free but we instead fix:
  - Location: $\beta_0 = 0$,
  - Scale: $\beta_1 = 1$.

Discuss

Why are these normalization requirements a good idea?

Estimation by OLS

- **Model:**
$$G(\mathbf{x}_i'\boldsymbol{\beta}_o) = \mathbf{x}_i\boldsymbol{\beta}_o.$$
- **Characterization/identification:** Since $\mathbb{E}(y_i|\mathbf{x}_i) = G(\mathbf{x}_i\boldsymbol{\beta}_o)$ (by ass.), follows from NLS proof that
$$\beta_o = \arg\min_{\beta} \mathbb{E}(y - \mathbf{x}\beta)^2.$$
- **Estimation:** closed-form solution [phew]
$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$
- **Question:** Where can predictions lie?
  - **Answer:**
- **Question:** What about implications for the error term variance $(\varepsilon_i \equiv y_i - G(\mathbf{x}_i\boldsymbol{\beta}_o))$.
  - **Answer:** The error term will be heteroskedastic.

Comparison