

Transition Models

t'_{ij} s are assumed to be equally spaced.

Let $H_i = \{y_k, k = 1, \dots, j-1\}$.

Consider

$$f(y_{ij} | H_{ij}, \alpha, \beta) = \exp \left\{ \frac{y_{ij} - \psi(\theta_{ij})}{\phi} + c(y_{ij}, \phi) \right\},$$

where $\psi(\theta_{ij})$ and $c(y_{ij}, \phi)$ are known functions.

One has

$$\mu_{ij}^c = E[y_{ij} | H_{ij}] = \psi'(\theta_{ij})$$

and

$$v_{ij}^c = V[y_{ij} | H_{ij}] = \psi''(\theta_{ij}) \phi$$

with

$$h(\mu_{ij}^c) = x_{ij}^T \beta + \sum_{r=1}^s f_r(H_{ij}; \alpha) \text{ for suitable functions } f_r(\cdot)'s,$$

and

$$v_{ij}^c = v(\mu_{ij}^c) \phi.$$

Problem 1. Fitting transition models: (A markov model of order q)

By

$$L_i(y_{i1}, \dots, y_{im_i}) = f(y_{i1}, \dots, y_{iq}) \prod_{j=q+1}^{m_i} f(y_{ij} | y_{i,j-1}, \dots, y_{i,j-q}), i = 1, \dots, n,$$

one can get the likelihood function

$$L(\alpha, \beta) = \prod_{i=1}^n f(y_{i1}, \dots, y_{iq}) \prod_{j=q+1}^{m_i} f(y_{ij} | H_{ij}, \alpha, \beta),$$

where

$$H_{ij} = \{y_{i,j-1}, \dots, y_{i,j-q}\}.$$

Since the term $f(y_{i1}, \dots, y_{iq})$ is always unavailable, the estimators of (α, β) are obtained via maximizing the conditional likelihood

$$\prod_{i=1}^n \prod_{j=q+1}^{m_i} f(y_{ij} | H_{ij}, \alpha, \beta).$$

Let $\theta = (\alpha, \beta)$.

Show that the log-conditional likelihood or conditional score function has the form

$$S^c(\theta) = \sum_{i=1}^n \sum_{j=(q+1)}^{m_i} \frac{\partial \mu_{ij}^c}{\partial \theta} v_{ij}^{c-1}(y_{ij} - \mu_{ij}^c).$$

$$L^c(\theta) = \prod_{i=1}^n \prod_{j=q+1}^{m_i} f(y_{ij} \mid H_{ij}, \alpha, \beta).$$

$$l^c(\theta) = \ln L^c(\theta) = \frac{\sum_{i=1}^n \sum_{j=(q+1)}^{m_i} (y_{ij} \theta_{ij} - \psi(\theta_{ij}))}{\phi} + \sum_{i=1}^n \sum_{j=(q+1)}^{m_i} c(y_{ij}, \phi).$$

We have

$$\begin{aligned} S^c(\theta) &= \frac{\partial l^c(\theta)}{\partial \theta} = \frac{\sum_{i=1}^n \sum_{j=(q+1)}^{m_i} (y_{ij} - \psi'(\theta_{ij}))}{\phi} \\ &= \sum_{i=1}^n \sum_{j=(q+1)}^{m_i} \frac{1}{\phi} (y_{ij} - \mu_{ij}^c) \\ &= \sum_{i=1}^n \sum_{j=(q+1)}^{m_i} \frac{\partial \mu_{ij}^c}{\partial \theta} v_{ij}^{c-1} (y_{ij} - \mu_{ij}^c), \end{aligned}$$

where

$$\begin{aligned} \mathbb{E}[y_{ij} \mid H_{ij}] &= \psi'(\theta_{ij}) \triangleq \mu_{ij}^c, \\ \mathbf{V}[y_{ij} \mid H_{ij}] &= \psi''(\theta_{ij})\phi = \frac{\partial \mu_{ij}^c}{\partial \theta} \phi \triangleq v_{ij}^c \implies \frac{1}{\phi} = \frac{\partial \mu_{ij}^c}{\partial \theta} v_{ij}^{c-1}. \end{aligned}$$

Problem 2. Ordered Categorical data

Y : ordinal response with categories labeled $1, 2, \dots, k$.

Let

$$F(a | x) = P(Y \leq a | x),$$

where $a = 1, \dots, (k-1)$, $x = (x_1, \dots, x_p)^T$.

Proportional odds model:

$$\text{logit } F(a | x) = \theta_a + x^T \beta, \quad a = 1, \dots, (k-1).$$

Define $Y^* = (Y_1^*, \dots, Y_{k-1}^*)$ with $Y_a^* = 1_{(Y \leq a)}$.

Then,

$$\text{logit } F(a | x) = \text{logit } P(Y_a^* = 1 | x).$$

Y	1	2	3	\dots	$k-1$	k
Y_1^*	1	0	0	\dots	0	0
Y_2^*	1	1	0	\dots	0	0
\vdots	\vdots	\vdots				\vdots
Y_{k-1}^*	1	1	1	\dots	1	0

Example:

Assume that

$$\text{logit } P(Y_j \leq b | Y_{i,j-1} = a) = \theta_{ab} + x_i^T \beta_a, \quad a, b = 1, \dots, (k-1).$$

It can be derived that

$$\text{logit } P(Y_{ij} \leq b | Y_{i,j-1}^* = y_{i,j-1}^*) = \theta_b + \sum_{l=1}^{k-1} \alpha_{lb} y_{i(j-1)l}^* + x_{ij}^T (\beta + \sum_{l=1}^{k-1} r_l y_{i(j-1)l}^*),$$

$$\text{where } \begin{cases} \theta_{kb} = \theta_b, \\ \alpha_{lb} = \theta_{lb} - \theta_{l+1b}, \\ \beta_k = \beta, \\ r_l = \beta_l - \beta_{l+1} \end{cases}.$$

$$Y_{ij} = a$$

$$\Rightarrow Y_{ij}^* = (\underbrace{0 \cdots 0}_{a-1} \underbrace{1 \cdots 1}_{k-a}).$$

$$\begin{aligned} \text{logit } P(Y_{ij} \leq b \mid Y_{ij-1}^* = y_{ij-1}^*) &= \theta_b + (\theta_{1b} - \theta_{2b})y_{i(j-1)1}^* + x_{ij}^T \{ \beta + (\beta_1 - \beta_2)y_{i(j-1)1}^* \\ &\quad + (\theta_{2b} - \theta_{3b})y_{i(j-1)2}^* + (\beta_2 - \beta_3)y_{i(j-1)2}^* \\ &\quad + \cdots + (\beta_{a-1} - \beta_a)y_{i(j-1)(a-1)}^* \\ &\quad + (\theta_{ab} - \theta_{a+1b})y_{i(j-1)a}^* + (\beta_a - \beta_{a+1})y_{i(j-1)a}^* \\ &\quad + (\theta_{a+1b} - \theta_{a+2b})y_{i(j-1)(a+1)}^* + (\beta_{a+1} - \beta_{a+2})y_{i(j-1)(a+1)}^* \\ &\quad + \cdots + (\theta_{(k-1)b} - \theta_{kb})y_{i(j-1)(k-1)}^* + (\beta_{k-1} - \beta_k)y_{i(j-1)(k-1)}^* \}. \end{aligned}$$

$$\begin{aligned} \text{logit } P(Y_{ij} \leq b \mid Y_{ij-1} = a) &= \theta_b + (\theta_{ab} - \theta_{a+1b}) + x_{ij}^T \{ \beta + (\beta_a - \beta_{a+1}) \\ &\quad + (\theta_{a+1b} - \theta_{a+2b}) + (\beta_{a+1} - \beta_{a+2}) \\ &\quad + \cdots + (\theta_{(k-1)b} - \theta_{kb}) + (\beta_{k-1} - \beta_k) \}. \end{aligned}$$

$$\begin{aligned} \text{logit } P(Y_{ij} \leq b \mid Y_{ij-1} = a) &= \theta_b + \theta_{ab} - \theta_{kb} + x_{ij}^T \{ \beta + \beta_a - \beta_k \} \\ &= \theta_{ab} + x_{ij}^T \beta_a. \end{aligned}$$

Problem 3. Log-linear transition models for count data

$Y_{ij} \mid (H_{ij}, x_{ij}) \sim \text{Poisson}(\mu_{ij}^c)$.

Model 1. Wong (1986) proposed that

$$\mu_{ij}^c = \exp(x_{ij}^T \beta) \{1 + \exp(-\alpha_0 - \alpha_1 y_{i,j-1})\},$$

$\alpha_0, \alpha_1 > 0$, where β is the influence of x_{ij} as $y_{i,j-1} = 0$.

Remark. When $y_{i,j-1} > 0$, μ_{ij}^c decreases as $y_{i,j-1}$ increases. A negative association is implied between the prior and current responses.

Model 2. $\mu_{ij}^c = \exp(x_{ij}^T \beta + \alpha y_{i,j-1})$.

Properties:

1. μ_{ij}^c increases as an exponential function of time as $\alpha > 0$.
2. When $\exp(x_{ij}^T \beta) = \mu$ and $\alpha < 0$, it leads to a stationary process.

$$\mathbb{E}[Y_{ij} \mid (H_{ij}, x_{ij})] = \text{Var}[Y_{ij} \mid (H_{ij}, x_{ij})] = \mu_{ij}^c.$$

We can obtain

$$\begin{aligned} \mathbb{E}[Y_{ij}] &= \mathbb{E}[\mu_{ij}^c] = \mathbb{E}[\exp(x_{ij}^T \beta + \alpha y_{i,j-1})] \\ &= \exp(x_{ij}^T \beta) \mathbb{E}[\exp(\alpha y_{i,j-1})] \\ &= \mu \mathbb{E}[\exp(\alpha y_{i,j-1})] \end{aligned}$$

If stationary,

$$\mu = \mu \exp(\alpha \mu)$$

Problem 4.

Model 3.

$$\mu_{ij} = \exp \left(x_{ij}^T \beta + \alpha \left\{ \ln \left(y_{ij-1}^* \right) - x_{ij-1}^T \beta \right\} \right),$$

where $y_{ij-1}^* = \max \{ y_{ij-1}, d \}$, $0 < d < 1$.

Property: $\begin{cases} \alpha = 0 : \text{it reduces to an ordinary log-linear model.} \\ \alpha < 0 : \text{negative correlation between } y_{ij-1} \text{ and } y_{ij} \\ \alpha > 0 : \text{positive correlation between } y_{ij-1} \text{ and } y_{ij} \end{cases}$

Application to a size-independent branching process:

$$\exp(x_{ij}^T \beta) = \mu$$

y_{ij} : the number of individuals in the i -th population at generation j

$Z_k(y_{ij-1})$: the number of offspring for person k in generation $(j-1)$

For $y_{ij-1} > 0$,

$$y_{ij} = \sum_{k=1}^{y_{ij-1}} Z_k(y_{ij-1}),$$

where

$$Z_k \stackrel{iid}{\sim} \text{Poisson} \left(\left(\frac{\mu}{y_{ij-1}^*} \right)^{1-\alpha} \right).$$

One can get

$$\mu_{ij}^c = \mu \cdot \left(\frac{y_{ij-1}}{\mu} \right)^\alpha.$$

Property:

- $\alpha < 0$: the sample paths oscillate back and forth about their long-term average level.
- $\alpha > 0$: the sample paths have sharper peaks and broader valleys.

R10A21126_HW05_Q4

November 12, 2023

```
[ ]: import numpy as np
import matplotlib.pyplot as plt

[ ]: # Function to calculate lamda for Poisson distribution
def calculate_lambda(yij_prev, mu, alpha):
    return (mu / yij_prev ) ** (1-alpha)

# Function to generate sample paths
def generate_sample_paths(num_generations, initial_values, mu, alpha):
    num_populations = len(initial_values)
    sample_paths = np.zeros((num_populations, num_generations))
    sample_paths[:, 0] = initial_values

    for j in range(1, num_generations):
        for i in range(num_populations):

            yij_prev = sample_paths[i, j-1]
            # print(yij_prev)

            Zk = np.random.poisson(calculate_lambda(yij_prev, mu[i], alpha),
size = round(yij_prev))

            yij = np.sum(Zk)
            sample_paths[i, j] = yij

    return sample_paths

def simulate_and_plot(num_generations, initial_values, mu, alpha):

    # Generate sample paths
    sample_paths = generate_sample_paths(num_generations, initial_values, mu,
alpha)

    # Plotting
    plt.figure(figsize=(10, 6))
    for i in range(len(initial_values)):
```

```

plt.plot(sample_paths[i, :], label=f'Population {i + 1}')
plt.axhline(y=np.average(sample_paths[i, :]), color='grey',
linestyle='--')

plt.title(f'Sample Paths for a = {alpha}')
plt.xlabel('Generation')
plt.ylabel('Population Size')
plt.legend(bbox_to_anchor=(1, 1))

```

```

[ ]: # Parameters
num_generations = 100 # Number of generations

# Set initial values of population
initial_values = np.array([800, 500, 300])
mu = initial_values

# Parameter alpha
alpha_values = [-0.1, -0.5, -.9]

for alpha in alpha_values:
    simulate_and_plot(num_generations, initial_values, mu, alpha)

plt.tight_layout()
plt.show()

```



