Transition Models

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

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

Consider

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

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

One has

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

and

$$V_{ij}^c = V\left[y_{ij} \mid H_{ij}\right] = \psi''\left(\theta_{ij}\right)\phi$$

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

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

one can get the likelihood function

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

where

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

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

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

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

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

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

Problem 2. Ordered Categorical data

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

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

where $a = 1, \cdots, (k-1), x = \left(x_1, \cdots, x_p\right)^T$.
Proportional odds model:

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

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

$$\text{logit}\, F(a \mid x) = \text{logit}\, P\left(Y_a^* = 1 \mid x\right).$$

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

....

Example:
Assume that

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

It can be derived that

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

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

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

$$
\begin{aligned}
&+(\beta_2-\beta_3)y^*_{i(j-1)2}\\
&+\cdots\\
&+(\beta_a-\beta_{a+1})y^*_{i(j-1)a}\\
&+\cdots\\
&+(\beta_{k-1}-\beta_k)y^*_{i(j-1)(k-1)}\},
\end{aligned}
$$

$$
\begin{aligned}
\operatorname{logit} P(Y_{ij}\le b\mid Y_{i\,j-1}=a) = {} & \theta_b+(\theta_{1b}-\theta_{2\,b})\\
&+(\theta_{2b}-\theta_{3\,b})\\
&+\cdots\\
&+(\theta_{ab}-\theta_{a+1\,b})\\
&+x_{ij}^T\{\beta\\
&+(\beta_1-\beta_2)\\
&+(\beta_2-\beta_3)\\
&+\cdots\\
&+(\beta_a-\beta_{a+1})\},\\
= {} & \theta_b+\theta_{1b}-\theta_{a+1\,b}+x_{ij}^T\{\beta+\beta_1-\beta_{a+1}\}.
\end{aligned}
$$

Problem 3. Log-linear transition models for count data

$Y_{ij} \mid \left(H_{ij}, x_{ij}\right) \sim$ Poissom $\left(\mu_{ij}^c\right)$.

....

Model 1. Wong (1986) proposed that

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

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

—

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

....

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

—

Properties:

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

....

Problem 4.

Moded 3.

$$\mu_{ij} = \exp\left({x_{ij}}^T \beta + \alpha \left\{ \ln\left(y^*_{i\,j-1} \right) - x^T_{i\,j-1}\beta \right\} \right),$$

where $y^*_{i\,j-1} = \max\left\{ y_{i\,j-1}, d \right\}, 0 < d < 1$.

—

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

—

Application to a size-independent branching process:
$\exp(x_{ij}^T\beta) = \mu$

$$\begin{aligned} y_{ij} &: \text{the number of individuals in the } i\text{-th population at generation } j \\ Z_k(y_{i\,j-1}) &: \text{the number of offspring for person } k \text{ in generation } (j-1) \end{aligned}$$

For $y_{i\,j-1} > 0$,

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

where

$$Z_k \overset{iid}{\sim} \mathit{Poisson}\left(\left(\frac{\mu}{y^*_{ij-1}} \right)^{1-\alpha} \right).$$

One can get

$$\mu^c_{ij} = \mu \cdot \left(\frac{y_{i(j-1)}}{\mu} \right)^{\alpha}.$$

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

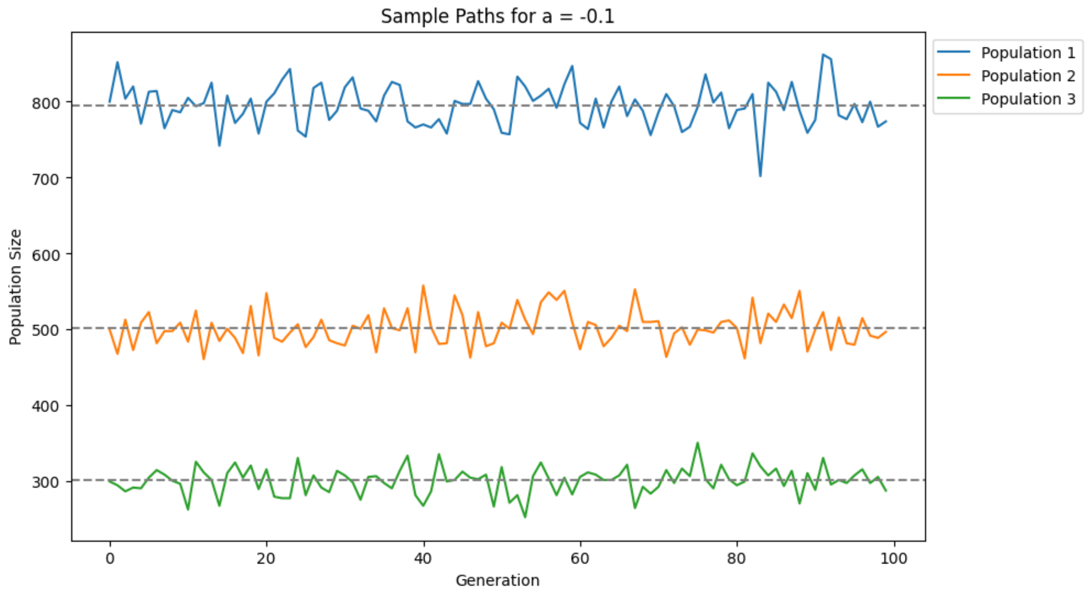
November 12, 2023

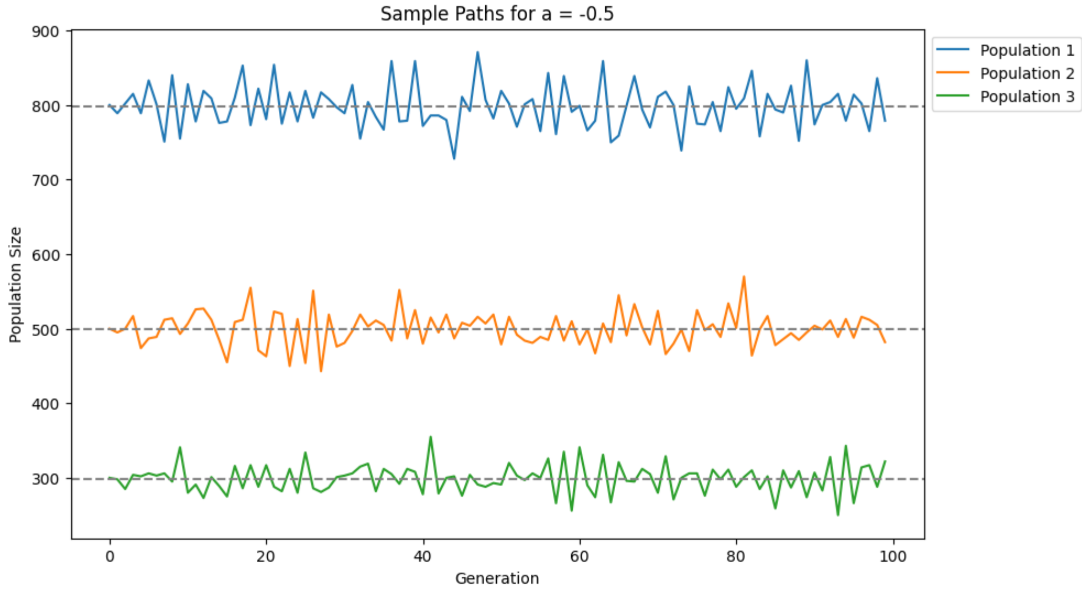
[ ]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

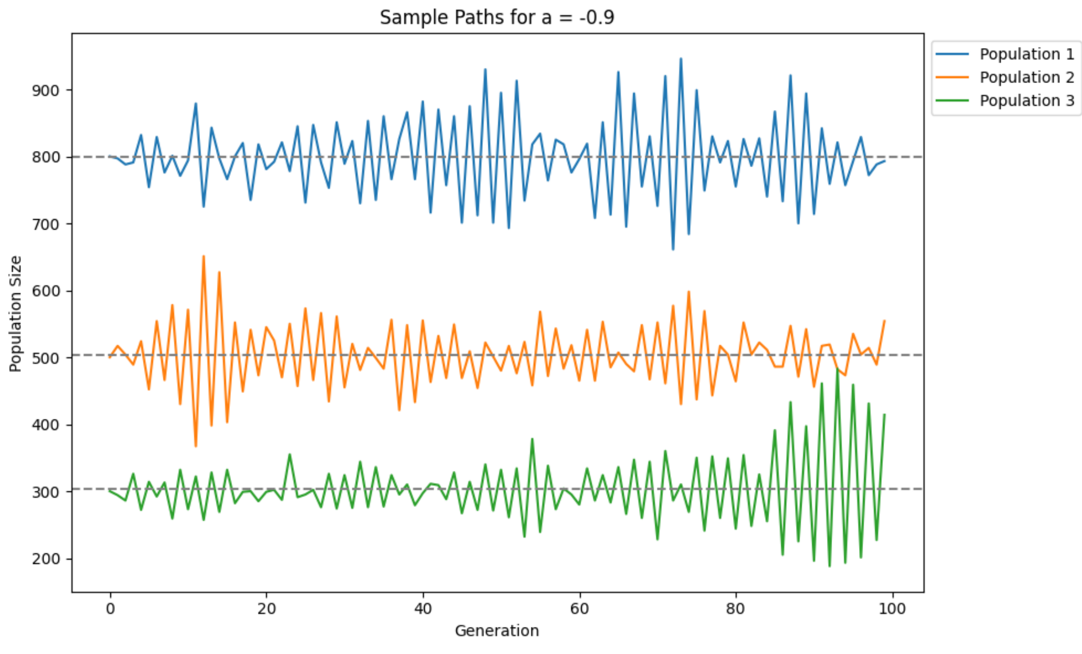
[ ]:
```python
# Function to calculate lamda for Poisson distribution
def calculate_lamda(yij_prev, mu, alpha):
    return (mu /yij_prev ) ** (1-alpha)

# 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_lamda(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()
```




Sample Paths for a = -0.5
Population 1
Population 2
Population 3
Population Size
Generation
900
800
700
600
500
400
300
0
20
40
60
80
100


Sample Paths for a = -0.9
Population 1
Population 2
Population 3
Population Size
Generation
900
800
700
600
500
400
300
200
0
20
40
60
80
100