

Ekonometryczne modele nieliniowe

Wykład 4

NMНК, MNW, metody
gradientowe

Literatura

- W. Greene (2012) *Econometric Analysis*, rozdz. 7.2 (str. 222-242)
- J. Hamilton (1994) *Time Series Analysis*, str. 133 – 151
- Chung-Ming Kuan (2007) *Introduction to Econometric Theory*, Institute of Economics, Academia Sinica, rozdział 8
– *do znalezienia w internecie*

Nieliniowa metoda najmniejszych kwadratów

- Model regresji nieliniowej (l zmiennych, k parametrów): $y = f(x; \beta) + e(\beta)$

- Przykład (1): $y_t = \alpha + \beta \frac{x_t^\gamma - 1}{\gamma} + \varepsilon_t$

- Przykład (2): $y_t = \begin{cases} \alpha + \beta x_t + \varepsilon_t & z_t > \gamma \\ \delta + \phi x_t^2 + \varepsilon_t & z_t \leq \gamma \end{cases}$

Interpretacje ekonomiczne

- Wpływ krańcowej zmiany x na y nie zawsze równy wartości parametru

$$\frac{\partial f(x_t; \beta)}{\partial x_{ti}}$$

Estymator NMNK

- Estymator minimalizuje sumę kwadratów reszt:

$$Q_T(\beta) = \frac{1}{T} \sum_{t=1}^T [y_t - f(x_t; \beta)]^2$$

- Warunek pierwszego rzędu:

$$\nabla_{\beta} Q_T(\beta) = -\frac{2}{T} \nabla_{\beta} f(x_1, \dots, x_T; \beta) [\mathbf{y} - f(x_1, \dots, x_T; \beta)] = \mathbf{0}_{k \times 1}$$

gdzie:

$$\nabla_{\beta} f(x_1, \dots, x_T; \beta) = \left[\nabla_{\beta} f(x_1; \beta) \quad \nabla_{\beta} f(x_2; \beta) \quad \dots \quad \nabla_{\beta} f(x_T; \beta) \right]_{k \times T}$$

Założenia NMNK

1. Warunkowa średnia dla y wynosi:
 $E(y | \mathbf{x}_i) = f(\mathbf{x}_i; \beta)$, a $f(\mathbf{x}_i; \beta)$ różniczkowalna
2. Identyfikowalność parametrów: nie istnieje $\beta_0 \neq \beta$ takie, że $\forall_{\mathbf{x}_i} f(\mathbf{x}_i; \beta_0) = f(\mathbf{x}_i; \beta)$
3. Składnik losowy: $E(\varepsilon_i | f(\mathbf{x}_i; \beta)) = 0$
4. $E(\varepsilon_i^2 | f(\mathbf{x}_i; \beta)) = \sigma^2 = \text{const} < \infty$
 $E(\varepsilon_i \varepsilon_j | f(\mathbf{x}_i; \beta), f(\mathbf{x}_j; \beta)) = 0$ dla $i \neq j$

Założenia NMNK

5. 1. i 2. moment \mathbf{x}_i z próby dążą do stałych z populacji, a \mathbf{x}_i ściśle egzogeniczny wobec ε_i
6. Istnieje dobrze zdefiniowany rozkład prawdopodobieństwa dla ε_i ,

$\varepsilon_i \sim \text{uncorrelated identically distributed}(0, \sigma^2 \mid f(\mathbf{x}_i; \beta))$

Założenia NMNK

- Jeśli można policzyć 2. pochodne $f(x; \beta)$ względem parametrów dla danych obserwacji $(x \text{ i } y)$ i macierz $\nabla_{\beta}^2 Q_T(\beta)$ jest dodatnio określona, to minimum funkcji (sumy kwadratów reszt) można znaleźć.
- Możliwe wiele minimów lokalnych tej funkcji □ parametry niekoniecznie „jednoznacznie” identyfikowalne (por. założenie 2)

Założenia NMNK

- Powyższe założenie analogiczne do założenia nr 1 w MNK (por. wykład 1):

$$\nabla_{\beta}^2 Q_T(\beta) = -\frac{2}{T} \nabla_{\beta}^2 \mathbf{f}(\beta) [\mathbf{y} - \mathbf{f}(\beta)] + \frac{2}{T} [\nabla_{\beta} \mathbf{f}(\beta)] [\nabla_{\beta} \mathbf{f}(\beta)]'$$

ponieważ dla modelu liniowego:

$$\nabla_{\beta}^2 Q_T(\beta) = \frac{2}{T} (\mathbf{X}' \mathbf{X}) \quad \square \quad rz(\mathbf{X}' \mathbf{X}) = k$$

Własności estymatorów NMNK

- Zgodność
- Asymptotyczna normalność

$$\sqrt{T}(\hat{\beta}_T - \beta) \xrightarrow{D} N(0; \hat{D}_T)$$

$$\hat{D}_T = \hat{\sigma}_T^2 \left(\frac{1}{T} \sum_{t=1}^T \left(\nabla f_{\beta}(x_t; \hat{\beta}_T) \right) \left(\nabla f_{\beta}(x_t; \hat{\beta}_T) \right)' \right)^{-1}$$

$$\hat{\sigma}_T^2 = \sum_{t=1}^T \frac{\hat{e}_t^2}{T}$$

Własności estymatorów NMNK

- Estymator wariancji odporny na heteroskedastyczność:

$$\hat{D}_T = \left(\frac{1}{T} \sum_{t=1}^T (\nabla f_{\beta}(x_t; \hat{\beta}_T)) (\nabla f_{\beta}(x_t; \hat{\beta}_T))' \right)^{-1} \hat{V}_T \left(\frac{1}{T} \sum_{t=1}^T (\nabla f_{\beta}(x_t; \hat{\beta}_T)) (\nabla f_{\beta}(x_t; \hat{\beta}_T))' \right)^{-1}$$
$$\hat{V}_T = \frac{4}{T} \sum_{t=1}^T \hat{\varepsilon}_t^2 (\nabla f_{\beta}(x_t; \hat{\beta}_T)) (\nabla f_{\beta}(x_t; \hat{\beta}_T))'$$

– możliwe też estymatory typu Neweya-Westa

- Test Walda – analogiczny jak dla MNK:

$$W = T(\mathbf{Rb} - \mathbf{r})(\mathbf{R}\hat{D}_T\mathbf{R}')^{-1} (\mathbf{Rb} - \mathbf{r})' \stackrel{a}{\sim} \chi^2(m)$$

Analogia do (*quasi*-) MNW

- Estymator m. kowariancji dla MNW:

$$\hat{D} = E(\hat{\theta} - \theta)(\hat{\theta} - \theta)' \approx T^{-1} (H_{2D} H_{OP}^{-1} H_{2D})^{-1}$$

gdzie:

$$H_{2D} = -T^{-1} \frac{\partial^2 L(\theta)}{\partial \theta \partial \theta'} \Big|_{\theta = \hat{\theta}}$$

$$L(\theta) = \sum_{t=1}^T \ln g(y_t | \Omega_{t-1}, \theta)$$

$$H_{OP} = T^{-1} \sum_{t=1}^T [h(\hat{\theta}, y_t) h(\hat{\theta}, y_t)']$$

$$h(\hat{\theta}) = \frac{\partial \ln g(y_t | \Omega_{t-1}, \theta)}{\partial \theta} \Big|_{\theta = \hat{\theta}}$$

Algorytmy dla NMNK i MNW

- Grid search
 - dobre wyniki, gdy poszukiwana wartość jednego parametru
 - przydatna jako część innych metod

Algorytmy dla NMNK i MNW

- Metoda „steepest ascent” (albo steepest descent) – najszybszego wzrostu (spadku)

– wartości startowe wektora parametrów θ : θ_0

– ustalamy długość kroku przy szukaniu optimum funkcji:

$$(\theta^1 - \theta^0)'(\theta^1 - \theta^0) = k$$

– szukamy maksimum przy warunkach:

$$\max_{\theta^1} L(\theta^1) \quad p.w. \quad (\theta^1 - \theta^0)'(\theta^1 - \theta^0) = k$$

Algorytmy dla NMNK i MNW

– Langrange'an ma postać:

$$J(\theta^1) = L(\theta^1) + \lambda(k - (\theta^1 - \theta^0)'(\theta^1 - \theta^0))$$

– Przyrównujemy pochodną po wektorze parametrów do zera:

$$\left. \frac{\partial L(\theta)}{\partial \theta} \right|_{\theta=\theta^1} - 2\lambda(\theta^1 - \theta^0) = 0$$

– Niech $g(\theta)$ oznacza gradient (logarytmu) funkcji wiarygodności po parametrach, to wtedy:

$$\theta^1 - \theta^0 = g(\theta^1) \cdot \frac{1}{2\lambda}$$

Algorytmy dla NMNK i MNW

– Jeżeli $k \rightarrow 0$, to $g(\theta^1) \rightarrow g(\theta^0)$, czyli:

$$\theta^1 - \theta^0 \approx g(\theta^0) \cdot s$$

– Kolejne kroki: $\theta^{m+1} - \theta^m = g(\theta^m) \cdot s$

– s może być wybrane przy pomocy metody „grid search”, tak by maksymalizować wartość $L(\theta)$

Algorytmy dla NMNK i MNW

- Można wyprowadzić długość kroku i wzór na iteracje przyjmie postać:

$$\theta^{m+1} - \theta^m = \frac{g(\theta^m)' g(\theta^m)}{g(\theta^m)' H(\theta^m) g(\theta^m)} g(\theta^m)$$

- Problemy:

- wiele maksimów lokalnych – wypróbuj wiele wartości startowych
- jeśli hessian H nie jest dodatnio określoną macierzą to algorytm może wskazywać przybliżenia w złym kierunku

Algorytmy dla NMNK i MNW

- Metoda Newtona-Raphsona
 - Szybsza zwykle niż metoda „najszybszego spadku” jeśli spełnione są warunki:
 - istnieją drugie pochodne funkcji $L(\theta)$,
 - funkcja $L(\theta)$ jest wypukła, tzn. $H(\theta)$ jest macierzą dodatnio określoną na całej przestrzeni parametrów

$$H(\theta^0) = - \left. \frac{\partial^2 L(\theta)}{\partial \theta \partial \theta'} \right|_{\theta = \theta^0}$$

Algorytmy dla NMNK i MNW

- przybliżenie logarytmu funkcji wiarygodności przy pomocy szeregu Taylora:

$$L(\theta) \approx L(\theta^0) + g(\theta^0)'(\theta - \theta^0) - \frac{1}{2}(\theta - \theta^0)'H(\theta^0)(\theta - \theta^0)$$

- przyrównujemy pochodną po wektorze parametrów do zera: $g(\theta^0) - H(\theta^0)(\theta - \theta^0) \approx 0$

- wyprowadzenie algorytmu:

$$g(\theta^0) = H(\theta^0)(\theta - \theta^0) \quad \theta^1 - \theta^0 = [H(\theta^0)]^{-1} g(\theta^0)$$

$$\theta^{m+1} = \theta^m + [H(\theta^m)]^{-1} g(\theta^m)$$

- często stosuje się kroki o różnej długości:

$$\theta^{m+1} = \theta^m + s^m \cdot [H(\theta^m)]^{-1} g(\theta^m)$$

Algorytmy dla NMNK i MNW

- Metoda Gaussa-Newtona (tylko dla NMNK)

$$\theta^{m+1} = \theta^m + \left[\left(\nabla_{\theta} f(\theta^m) \right)' \left(\nabla_{\theta} f(\theta^m) \right) \right]^{-1} \left(\nabla_{\theta} f(\theta^m) \right) [y - f(\theta^m)]$$

- gdzie $\left(\nabla_{\theta} f(\theta^m) \right)$ oznacza gradient funkcji f po parametrach
- To przybliżenie macierzy H jest dodatnio określone.
- Można zastosować MNK do wyznaczenia kroku!

Algorytmy dla NMNKK i MNW

- Korekty na dodatnią określoność macierzy H

$$\left[H(\theta^m) \right]_c^{-1} = \left[H(\theta^m) \right]^{-1} + cI$$

używana w algorytmie
„Marquardt-Levenberg” ($c > 0$)

$$\left[H(\theta^{m+1}) \right]^{-1} = \left[H(\theta^m) \right]^{-1} + C^m$$

metoda quasi-Newtona
 $\left[H(\theta^0) \right]^{-1} = I$

Algorytmy dla NMNK i MNW

- Davidon-Fletcher-Powell
 - ułatwienie - nie trzeba liczyć hessianu w każdym kroku (jego wartość jest jedynie przybliżana)

$$\theta^{m+1} = \theta^m + s \cdot A^m g(\theta^m)$$

$$A^{m+1} = A^m - \frac{A^m (\Delta g^{m+1})(\Delta g^{m+1})' A^m}{(\Delta g^{m+1})' A^m (\Delta g^{m+1})} - \frac{(\Delta \theta^{m+1})(\Delta \theta^{m+1})'}{(\Delta g^{m+1})' (\Delta \theta^{m+1})}$$

$$A^0 = I_k$$

$$\Delta g^{m+1} = g(\theta^{m+1}) - g(\theta^m), \quad \Delta \theta^{m+1} = \theta^{m+1} - \theta^m$$

Algorytmy dla NMNK i MNW

- Zakończenie algorytmu

$$\|\theta^{m+1} - \theta^m\| < c$$

$$\|g(\theta^m)\| < c$$

$$|L(\theta^{m+1}) - L(\theta^m)| < c$$

lub po przekroczeniu pewnej liczby kroków

Symulowane wyżarzanie -simulated annealing

- Wycięte z: Goffe, Ferrier, Rogers (1994)

SET initial parameters and values

set C (controls how fast V adjusts)
set X (starting values for model parameters)
set V (should cover the entire range of interest in X)
set ε (convergence criteria)
set r_T (temperature reduction factor)
set T_0 (initial temperature)
set N_ε (# times ε tolerance is achieved before termination)
set N_S (# times through function before V adjustment)
set N_T (# times through N_S loop before T reduction)

CALCULATE $f(X)$

$$X_{\text{opt}} = X$$

$$f_{\text{opt}} = f$$

Symulowane wyżarzanie (2)

```
DO UNTIL convergence
  DO  $N_T$  times
    DO  $N_S$  times
      DO  $i = 1, n$ 
         $x'_i = x_i + r \cdot v_i$  { $r$  is uniform random number on  $[-1, 1]$ }
        CALCULATE  $f'(X')$ 
        IF  $f' \leq f$  THEN
          apply Metropolis criteria
          IF accepted:  $X = X' \ \& \ f = f'$ 
        END IF
        IF  $f' > f$  THEN
           $X = X' \ \& \ f = f'$ 
        END IF
        IF  $f' > f_{\text{opt}}$  THEN
           $X = X', f = f', X_{\text{opt}} = X', \ \& \ f_{\text{opt}} = f'$ 
        END IF
      END DO
    END DO
    ADJUST  $V$  such that half of all trials are accepted
  END DO
```

Symulowane wyżarzanie (3)

```
IF change in  $f_{\text{opt}} < \varepsilon$  last  $N_\varepsilon$  iterations &  $|f - f'| < \varepsilon$  THEN  
  REPORT  $X_{\text{opt}}, f_{\text{opt}}, \& V$   
  STOP  
ELSE  
   $X = X_{\text{opt}}$  {start on current best optimum}  
   $T = r_T \cdot T$  {reduce  $T$ }  
END IF
```

CONTINUE

- Kryterium wyboru nowego punktu:

If f' is less than or equal to f , the Metropolis criterion decides on acceptance (thermodynamics motivates this criterion). The value

$$p = e^{(f' - f)/T} \tag{2}$$

is computed and compared to p' , a uniformly distributed random number from $[0, 1]$. If p is greater than p' , the new point is accepted, X is updated with X' , and the algorithm moves downhill. Otherwise, X' is rejected.

Algorytm Nelder-Mead

- Wycięte z wikipedii (jest też polska wersja językowa) - minimalizacja:

1. **Order** according to the values at the vertices:

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \dots \leq f(\mathbf{x}_{n+1})$$

2. Calculate \mathbf{x}_o , the center of gravity of all points except \mathbf{x}_{n+1} .

3. **Reflection**

Compute reflected point $\mathbf{x}_r = \mathbf{x}_o + \alpha(\mathbf{x}_o - \mathbf{x}_{n+1})$

If the reflected point is better than the second worst, but not better than the best, i.e.:

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_r) < f(\mathbf{x}_n),$$

then obtain a new simplex by replacing the worst point \mathbf{x}_{n+1} with the reflected point \mathbf{x}_r , and go to step 1.

Algorithm Nelder-Mead (2)

4. Expansion

If the reflected point is the best point so far, $f(\mathbf{x}_r) < f(\mathbf{x}_1)$,

then compute the expanded point $\mathbf{x}_e = \mathbf{x}_o + \gamma(\mathbf{x}_o - \mathbf{x}_{n+1})$

If the expanded point is better than the reflected point, $f(\mathbf{x}_e) < f(\mathbf{x}_r)$

then obtain a new simplex by replacing the worst point \mathbf{x}_{n+1} with the expanded point \mathbf{x}_e , and go to step 1.

Else obtain a new simplex by replacing the worst point \mathbf{x}_{n+1} with the reflected point \mathbf{x}_r , and go to step 1.

Else (i.e. reflected point is not better than second worst) continue at step 5.

Algorytm Nelder-Mead (3)

5. Contraction

Here, it is certain that $f(\mathbf{x}_r) \geq f(\mathbf{x}_n)$

Compute contracted point $\mathbf{x}_c = \mathbf{x}_o + \rho(\mathbf{x}_o - \mathbf{x}_{n+1})$

If the contracted point is better than the worst point, i.e. $f(\mathbf{x}_c) < f(\mathbf{x}_{n+1})$

then obtain a new simplex by replacing the worst point \mathbf{x}_{n+1} with the contracted point \mathbf{x}_c , and go to step 1.

Else go to step 6.

6. Reduction

For all but the best point, replace the point with

$x_i = x_1 + \sigma(x_i - x_1)$ for all $i \in \{2, \dots, n+1\}$. go to step 1.

- Standardowe wartości parametrów:

$$\alpha = 1, \gamma = 2, \rho = -1/2 \quad \sigma = 1/2.$$