LAMPIRAN

Lampiran 1

Asumsi-asumsi pada Teorema 2.5.1

Asumsi-asumsi pada Teorema 2.5.1 sebagai berikut:

B1 : Ruang parameter, Θ, merupakan suatu himpunan bagian yang terbuka dari R^p.

B2 : Himpunan $A = \{x : f(x;\boldsymbol{\theta}) > 0\}$ tidak tergantung pada $\boldsymbol{\theta}$.

B3 : $f(x;\boldsymbol{\theta})$ kontinu dan diferensiabel sebanyak tiga kali terhadap $\boldsymbol{\theta}$ untuk semua *x* di *A*.

B4 : $E_\theta[\ell'(X_i;\boldsymbol{\theta})] = 0$ untuk semua $\boldsymbol{\theta}$ dan $Cov_\theta[\ell'(X_i;\boldsymbol{\theta})] = I(\boldsymbol{\theta})$ dimana $I(\boldsymbol{\theta})$ positif definit untuk semua $\boldsymbol{\theta}$.

B5 : $E_\theta[\ell''(X_i;\boldsymbol{\theta})] = -J(\boldsymbol{\theta})$ dimana $J(\boldsymbol{\theta})$ positif definit untuk semua $\boldsymbol{\theta}$.

B6 : Misalkan $\ell'''_{jkl}(x;\boldsymbol{\theta})$ merupakan turunan parsial gabungan dari $\ell(x;\boldsymbol{\theta})$ terhadap $\theta_j, \theta_k, \theta_l$. Untuk setiap $\boldsymbol{\theta}$, $\delta > 0$ dan $1 \le j,k,l \le p$,

$$| \ell'''_{jkl}(x;\boldsymbol{t}) | \le M_{jkl}(x)$$

untuk $\|\boldsymbol{\theta} - \boldsymbol{t}\| \le \delta$ dimana $E_\theta[M_{jkl}(X_i)] < \infty$.

Lampiran 2

Pembuktian Teorema 2.5.1

Teorema

Misalkan $X_1, \ldots, X_n$ i.i.d variabel random dengan suatu densitas atau fungsi frekuensi yang memenuhi kondisi B1 sampai B6 dengan $I(\boldsymbol{\theta}) = J(\boldsymbol{\theta})$ dimana $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$. Jika MLE $\hat{\boldsymbol{\theta}}_n$ memenuhi $\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) \xrightarrow{d} N(0, I^{-1}(\boldsymbol{\theta}))$ maka LR statistik Λ_n untuk menguji $H_0 : \theta_1 = \theta_{10}, \ldots, \theta_r = \theta_{r0}$ memenuhi

$$2\ln(\Lambda_n) \xrightarrow{d} V \sim \chi_r^2$$

di bawah kondisi H_0 benar.

Bukti:

Misalkan $\ell(x;\boldsymbol{\theta}) = \ln f(x;\boldsymbol{\theta})$ dan $\ell'(x;\boldsymbol{\theta}), \ell''(x;\boldsymbol{\theta})$ menyatakan turunan terhadap $\boldsymbol{\theta}$. Di bawah kondisi B1 sampai B6 pada teorema,

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_{n0}) \xrightarrow{d} N(0, I^{-1}(\boldsymbol{\theta}_{n0}))$$

di bawah kondisi H_0 benar.

Dengan me-*log*-kan dan menggunakan ekspansi deret Taylor, diperoleh

$$\begin{aligned}\ln(\Lambda_n) &= \sum_{i=1}^{n}[\ell(X_i;\hat{\boldsymbol{\theta}}_n) - \ell(X_i;\hat{\boldsymbol{\theta}}_{n0})] \\ &= (\boldsymbol{\theta}_{n0} - \hat{\boldsymbol{\theta}}_n)^2 \sum_{i=1}^{n} \ell'(X_i;\hat{\boldsymbol{\theta}}_n) - \frac{1}{2}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_{n0})^2 \sum_{i=1}^{n} \ell''(X_i;\boldsymbol{\theta}_n^*) - \sum_{i=1}^{n} \ell(X_i;\hat{\boldsymbol{\theta}}_{n0}) \\ &= -\frac{1}{2} n(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_{n0})^2 \frac{1}{n}\sum_{i=1}^{n} \ell''(X_i;\boldsymbol{\theta}_n^*)\end{aligned}$$

dimana $\boldsymbol{\theta}_n^*$ diantara $\boldsymbol{\theta}_{n0}$ dan $\hat{\boldsymbol{\theta}}_n$.

Di bawah asumsi B5 dan B6, di bawah kondisi H_0 benar diperoleh

$$\frac{1}{n}\sum_{i=1}^{n} \ell''(X_i;\boldsymbol{\theta}_n^*) \xrightarrow{p} -E_{\boldsymbol{\theta}_{n0}}[\ell''(X_i;\boldsymbol{\theta}_{n0})] = I(\boldsymbol{\theta}_{n0}) = J(\boldsymbol{\theta}_{n0})$$

Karena

$$n(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_{n0}) \xrightarrow{d} \frac{V}{I(\boldsymbol{\theta}_{n0})}$$

kesimpulan diperoleh dengan menggunakan Teorema Slutsky.

Teorema Slutsky

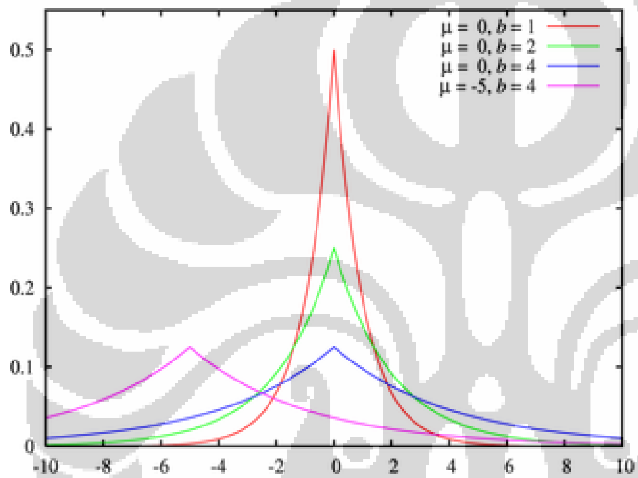
Misalkan $X_n \xrightarrow{d} X$ dan $Y_n \xrightarrow{p} \theta$ (suatu konstanta) , maka

$X_n Y_n \xrightarrow{d} \theta X$.

Lampiran 3

Penurunan Beberapa Distribusi untuk Memperoleh Pola Pemilihan Tilt Function

1. Distribusi Laplace



a. $$\frac{g_1(x)}{g_2(x)} = \frac{\frac{1}{2b_1}\exp\left\{-\frac{\mu_1 - x}{b_1}\right\}}{\frac{1}{2b_2}\exp\left\{-\frac{\mu_2 - x}{b_2}\right\}}$$

$$= \exp\left\{\ln\frac{b_2}{b_1} - \frac{\mu_1}{b_1} + \frac{\mu_2}{b_2} + x\left(\frac{1}{b_1} - \frac{1}{b_2}\right)\right\}$$

sehingga diperoleh

$$\alpha_1 = \ln\frac{b_2}{b_1} - \frac{\mu_1}{b_1} + \frac{\mu_2}{b_2}, \quad \beta_1 = \frac{1}{b_1} - \frac{1}{b_2}, \quad h(x) = x$$

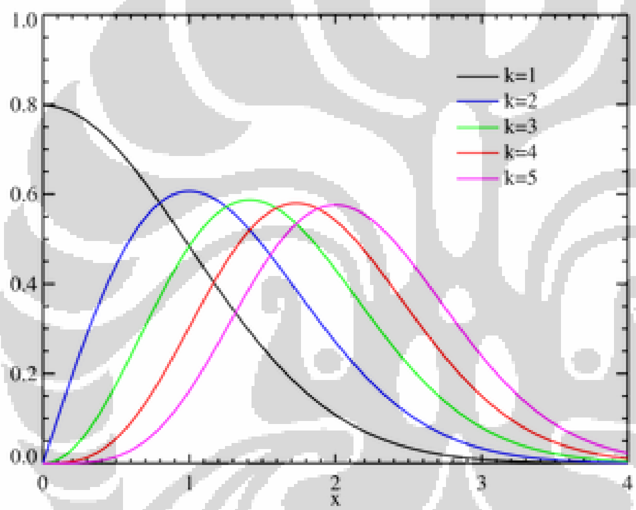
b. $$\frac{g_1(x)}{g_2(x)} = \frac{\frac{1}{2b_1}\exp\left\{-\frac{x - \mu_1}{b_1}\right\}}{\frac{1}{2b_2}\exp\left\{-\frac{x - \mu_2}{b_2}\right\}}$$

$$= \exp\left\{ \ln \frac{b_2}{b_1} + \frac{\mu_1}{b_1} - \frac{\mu_2}{b_2} + x\left(\frac{1}{b_2} - \frac{1}{b_1} \right) \right\}$$

sehingga diperoleh

$$\alpha_1 = \ln \frac{b_2}{b_1} + \frac{\mu_1}{b_1} - \frac{\mu_2}{b_2}, \quad \beta_1 = \frac{1}{b_2} - \frac{1}{b_1}, \quad h(x) = x$$

2. Distribusi Chi



$$\frac{g_1(x)}{g_2(x)} = \frac{\dfrac{x^{k_1-1}\exp\left(-\dfrac{x^2}{2}\right)}{2^{\frac{k_1}{2}-1}\Gamma\left(\dfrac{k_1}{2}\right)}}{\dfrac{x^{k_2-1}\exp\left(-\dfrac{x^2}{2}\right)}{2^{\frac{k_2}{2}-1}\Gamma\left(\dfrac{k_2}{2}\right)}}$$

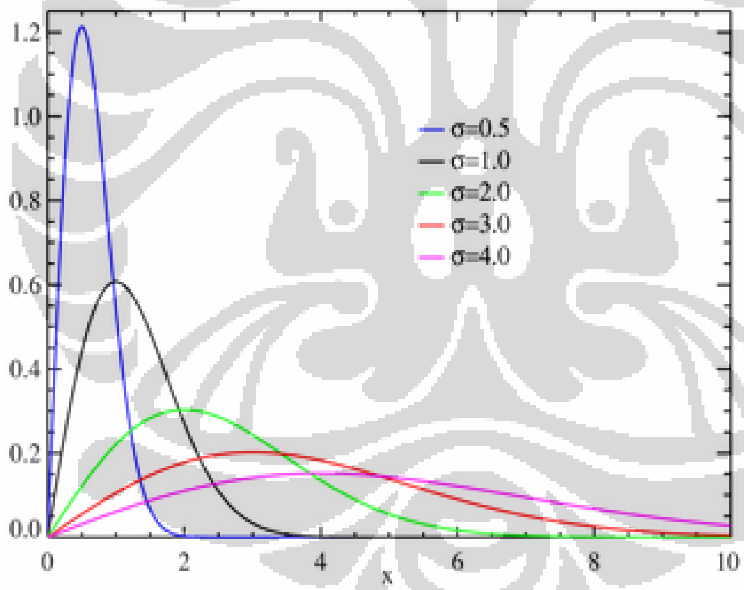
$$= \frac{x^{k_1} 2^{\frac{k_2}{2}-1}\Gamma\left(\dfrac{k_2}{2}\right)}{x^{k_2} 2^{\frac{k_1}{2}-1}\Gamma\left(\dfrac{k_1}{2}\right)}$$

$$= \exp\left\{(k_1 - k_2)\ln x + \ln\left(\frac{2^{\frac{k_2}{2}-1}\Gamma\left(\frac{k_2}{2}\right)}{2^{\frac{k_1}{2}-1}\Gamma\left(\frac{k_1}{2}\right)}\right)\right\}$$

sehingga diperoleh

$$\alpha_1 = \ln\left(\frac{2^{\frac{k_2}{2}-1}\Gamma\left(\frac{k_2}{2}\right)}{2^{\frac{k_1}{2}-1}\Gamma\left(\frac{k_1}{2}\right)}\right), \quad \beta_1 = k_1 - k_2, \quad h(x) = \ln x$$

3. Distribusi Rayleigh



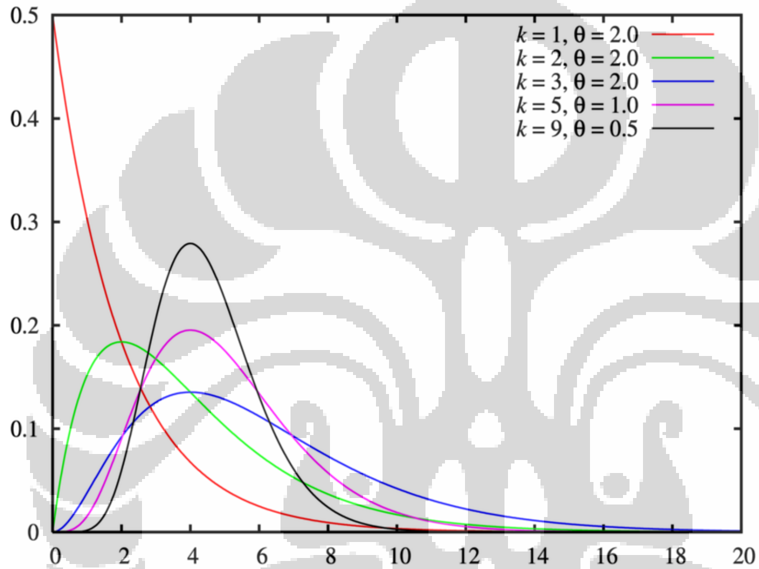
$$\frac{g_1(x)}{g_2(x)} = \frac{\dfrac{x\exp\left(-\dfrac{x^2}{2\sigma_1^2}\right)}{\sigma_1^2}}{\dfrac{x\exp\left(-\dfrac{x^2}{2\sigma_2^2}\right)}{\sigma_2^2}}$$

$$= \exp\left\{2\ln\frac{\sigma_2}{\sigma_1} + x^2\left(\frac{1}{2\sigma_2^2} - \frac{1}{2\sigma_1^2}\right)\right\}$$

sehingga diperoleh

$$\alpha_1 = 2\ln\frac{\sigma_2}{\sigma_1}\,, \quad \beta_1 = \frac{1}{2\sigma_2^2} - \frac{1}{2\sigma_1^2}\,, \quad h(x) = x^2$$

4. Distribusi Gamma



$$\frac{g_1(x)}{g_2(x)} = \frac{\dfrac{1}{\Gamma(\alpha_1)\beta_1^{\alpha_1}} x^{\alpha_1 - 1} \exp\left(-\dfrac{x}{\beta_1}\right)}{\dfrac{1}{\Gamma(\alpha_2)\beta_2^{\alpha_2}} x^{\alpha_2 - 1} \exp\left(-\dfrac{x}{\beta_2}\right)}$$

$$= \frac{\Gamma(\alpha_2)\beta_2^{\alpha_2} x^{\alpha_1} \exp\left(-\dfrac{x}{\beta_1}\right)}{\Gamma(\alpha_1)\beta_1^{\alpha_1} x^{\alpha_2} \exp\left(-\dfrac{x}{\beta_2}\right)}$$

$$= \exp\left\{ \ln \frac{\Gamma(\alpha_2)}{\Gamma(\alpha_1)} + \alpha_2 \ln \beta_2 - \alpha_1 \ln \beta_1 + \alpha_1 \ln x - \alpha_2 \ln x - \frac{x}{\beta_1} + \frac{x}{\beta_2} \right\}$$

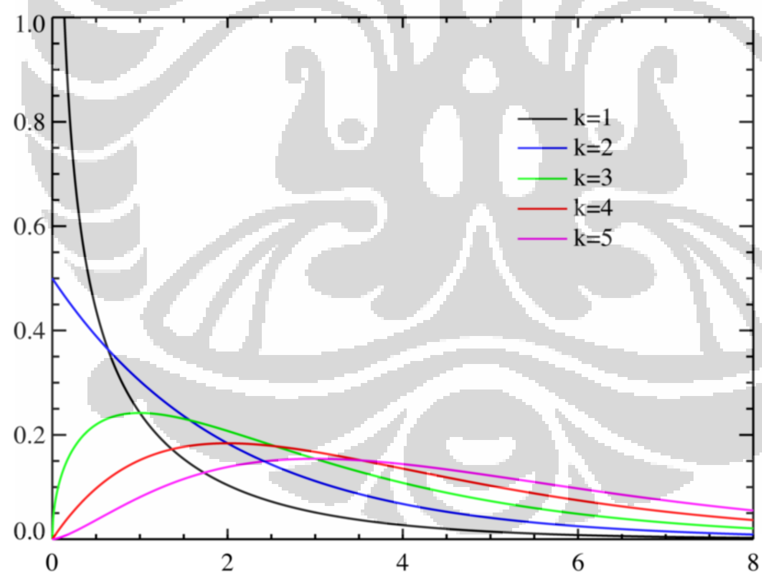
$$= \exp\left\{ \ln \frac{\Gamma(\alpha_2)}{\Gamma(\alpha_1)} + \alpha_2 \ln \beta_2 - \alpha_1 \ln \beta_1 + (\alpha_1 - \alpha_2) \ln x + x\left(\frac{1}{\beta_2} + \frac{1}{\beta_1} \right) \right\}$$

$$= \exp\left\{ \ln \frac{\Gamma(\alpha_2)}{\Gamma(\alpha_1)} + \alpha_2 \ln \beta_2 - \alpha_1 \ln \beta_1 + \begin{pmatrix} \alpha_1 - \alpha_2 \\ \dfrac{1}{\beta_2} + \dfrac{1}{\beta_1} \end{pmatrix} (\ln x \quad x) \right\}$$

sehingga diperoleh

$$\alpha = \ln \frac{\Gamma(\alpha_2)}{\Gamma(\alpha_1)} + \alpha_2 \ln \beta_2 - \alpha_1 \ln \beta_1, \quad \boldsymbol{\beta} = \begin{pmatrix} \alpha_1 - \alpha_2 \\ \dfrac{1}{\beta_2} + \dfrac{1}{\beta_1} \end{pmatrix}, \quad \boldsymbol{h}(x) = (\ln x,\ x)$$

5. Distribusi Chi-Square



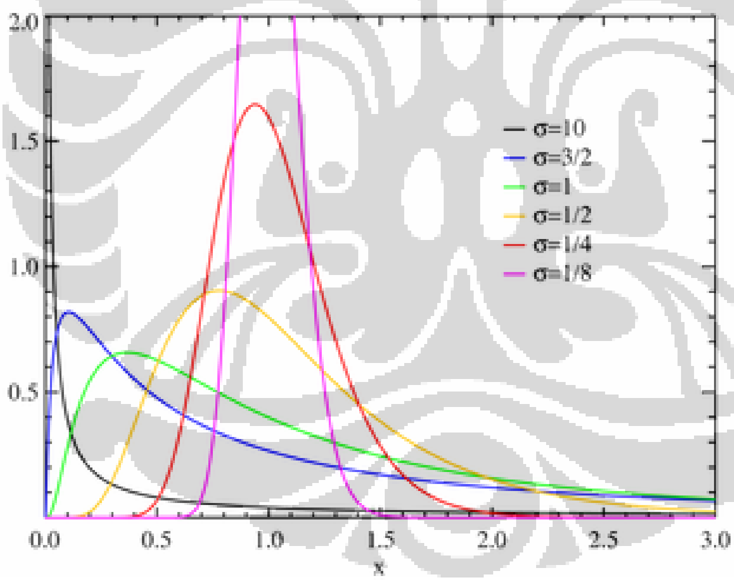
$$\frac{g_1(x)}{g_2(x)} = \frac{\dfrac{1}{\Gamma\left(\frac{r_1}{2}\right) 2^{\frac{r_1}{2}}} x^{\frac{r_1}{2}-1} \exp\left(-\dfrac{x}{2}\right)}{\dfrac{1}{\Gamma\left(\frac{r_2}{2}\right) 2^{\frac{r_2}{2}}} x^{\frac{r_2}{2}-1} \exp\left(-\dfrac{x}{2}\right)}$$

$$= \exp\left\{ \ln \frac{\Gamma\left(\frac{r_2}{2}\right) 2^{\frac{r_2}{2}}}{\Gamma\left(\frac{r_1}{2}\right) 2^{\frac{r_1}{2}}} + \ln x \left(\frac{r_1}{2} - \frac{r_2}{2} \right) \right\}$$

sehingga diperoleh

$$\alpha_1 = \ln \frac{\Gamma\left(\frac{r_2}{2}\right) 2^{\frac{r_2}{2}}}{\Gamma\left(\frac{r_1}{2}\right) 2^{\frac{r_1}{2}}}, \quad \beta_1 = \frac{r_1}{2} - \frac{r_2}{2}, \quad h(x) = \ln x$$

6. Distribusi Log-Normal



$$\frac{g_1(x)}{g_2(x)} = \frac{\frac{1}{x\sigma_1\sqrt{2\pi}} \exp\left\{ -\frac{(\ln x - \mu_1)^2}{2\sigma_1^2} \right\}}{\frac{1}{x\sigma_2\sqrt{2\pi}} \exp\left\{ -\frac{(\ln x - \mu_2)^2}{2\sigma_2^2} \right\}}$$

$$= \exp\left\{\ln\frac{\sigma_2}{\sigma_1} - \frac{\mu_1}{2\sigma_1^2} + \frac{\mu_2}{2\sigma_2^2} + \ln^2 x\left(\frac{1}{2\sigma_2^2} - \frac{1}{2\sigma_1^2}\right) + \ln x\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right)\right\}$$

$$= \exp\left\{\ln\frac{\sigma_2}{\sigma_1} - \frac{\mu_1}{2\sigma_1^2} + \frac{\mu_2}{2\sigma_2^2} + \begin{pmatrix} \frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2} \\ \frac{1}{2\sigma_2^2} - \frac{1}{2\sigma_1^2} \end{pmatrix} \left(\ln x \quad \ln^2 x\right)\right\}$$

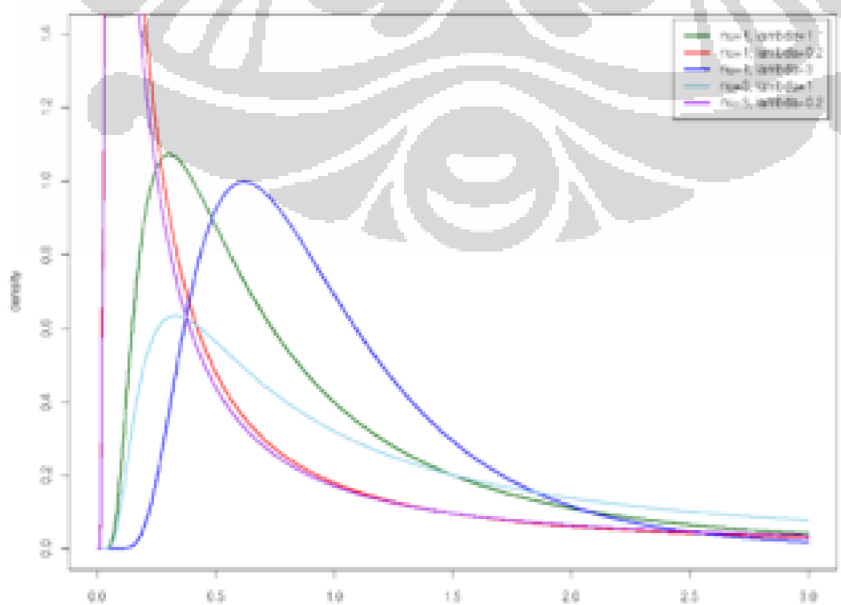
sehingga diperoleh

$$\alpha_1 = \ln\frac{\sigma_2}{\sigma_1} - \frac{\mu_1}{2\sigma_1^2} + \frac{\mu_2}{2\sigma_2^2},$$

$$\boldsymbol{\beta}_1 = \begin{pmatrix} \frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2} \\ \frac{1}{2\sigma_2^2} - \frac{1}{2\sigma_1^2} \end{pmatrix},$$

dan $\boldsymbol{h}(x) = \left(\ln x \quad \ln^2 x\right)$.

7. Distribusi Inverse Gaussian



$$\frac{g_1(x)}{g_2(x)} = \frac{\left[\dfrac{\lambda_1}{2\pi x^3}\right]^{1/2} \exp\left\{\dfrac{-\lambda_1(x^2 - 2x\mu_1 + \mu_1^2)^2}{2\mu_1^2 x}\right\}}{\left[\dfrac{\lambda_2}{2\pi x^3}\right]^{1/2} \exp\left\{\dfrac{-\lambda_2(x^2 - 2x\mu_2 + \mu_2^2)^2}{2\mu_2^2 x}\right\}}$$

$$= \exp\left\{\frac{1}{2}\ln\frac{\lambda_2}{\lambda_1} + \frac{\lambda_1}{2\mu_1} - \frac{\lambda_2}{2\mu_2} + x\left(\frac{\lambda_2}{2\mu_2^2} - \frac{\lambda_1}{2\mu_1^2}\right) + \frac{1}{x}\left(\frac{\lambda_2}{2} - \frac{\lambda_1}{2}\right)\right\}$$

$$= \exp\left\{\frac{1}{2}\ln\frac{\lambda_2}{\lambda_1} + \frac{\lambda_1}{2\mu_1} - \frac{\lambda_2}{2\mu_2} + \begin{pmatrix} \dfrac{\lambda_2}{2\mu_2^2} - \dfrac{\lambda_1}{2\mu_1^2} \\ \dfrac{\lambda_2}{2} - \dfrac{\lambda_1}{2} \end{pmatrix}\begin{pmatrix} x & \dfrac{1}{x} \end{pmatrix}\right\}$$

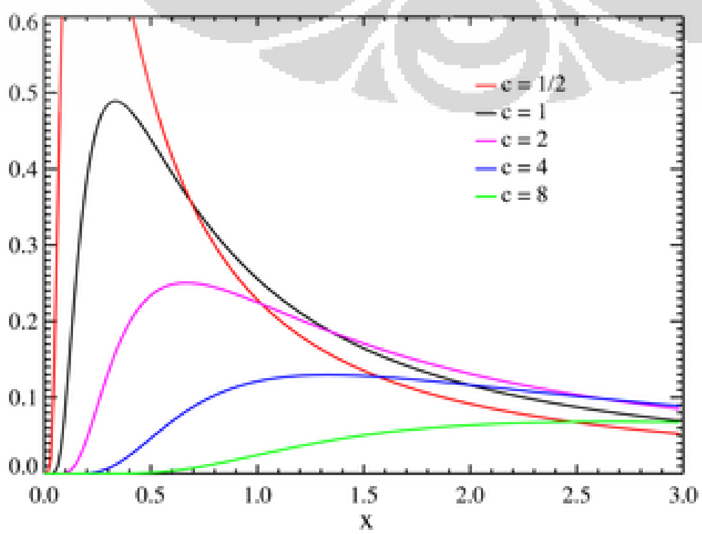
sehingga diperoleh

$$\alpha_1 = \frac{1}{2}\ln\frac{\lambda_2}{\lambda_1} + \frac{\lambda_1}{2\mu_1} - \frac{\lambda_2}{2\mu_2},$$

$$\boldsymbol{\beta}_1 = \begin{pmatrix} \dfrac{\lambda_2}{2\mu_2^2} - \dfrac{\lambda_1}{2\mu_1^2} \\ \dfrac{\lambda_2}{2} - \dfrac{\lambda_1}{2} \end{pmatrix},$$

dan $\boldsymbol{h}(x) = \begin{pmatrix} x & \dfrac{1}{x} \end{pmatrix}$.

8. Distribusi Levy



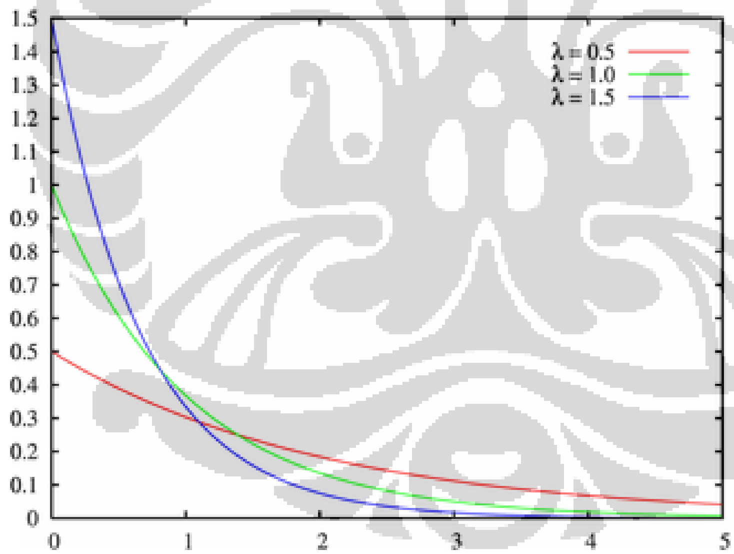
$$\frac{g_1(x)}{g_2(x)} = \frac{\sqrt{\frac{c_1}{2\pi}}\frac{\exp\left(-\frac{c_1}{2x}\right)}{x^{3/2}}}{\sqrt{\frac{c_2}{2\pi}}\frac{\exp\left(-\frac{c_2}{2x}\right)}{x^{3/2}}}$$

$$= \exp\left\{\frac{1}{2}\ln\frac{c_2}{c_1} + \frac{1}{x}\left(\frac{c_2}{2} - \frac{c_1}{2}\right)\right\}$$

sehingga diperoleh

$$\alpha_1 = \frac{1}{2}\ln\frac{c_2}{c_1},\quad \beta_1 = \frac{c_2}{2} - \frac{c_1}{2},\quad h(x) = \frac{1}{x}$$

9. Distribusi Eksponensial



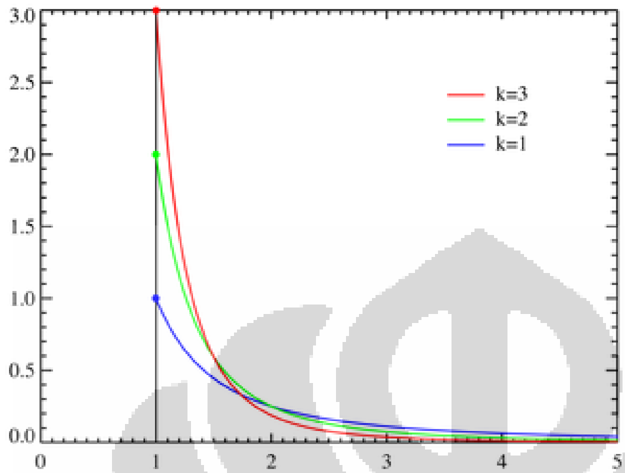
$$\frac{g_1(x)}{g_2(x)} = \frac{\lambda_1 \exp(-\lambda_1 x)}{\lambda_2 \exp(-\lambda_2 x)}$$

$$= \exp\left\{\ln\frac{\lambda_1}{\lambda_2} + (\lambda_2 - \lambda_1)x\right\}$$

sehingga diperoleh

$$\alpha_1 = \ln\frac{\lambda_1}{\lambda_2},\quad \beta_1 = \lambda_2 - \lambda_1,\quad h(x) = x$$

10. Distribusi Pareto



$$\frac{g_1(x)}{g_2(x)} = \frac{\dfrac{k_1 x_{m_1}^{k_1}}{x^{k_1+1}}}{\dfrac{k_2 x_{m_2}^{k_2}}{x^{k_2+1}}}$$

$$= \frac{k_1}{k_2} x^{k_2-k_1} x_{m_1}^{k_1} x_{m_2}^{-k_2}$$

$$= \exp\left\{ \ln\frac{k_1}{k_2} + k_1 \ln x_{m_1} - k_2 \ln x_{m_2} + (k_2 - k_1)\ln x \right\}$$

sehingga diperoleh

$$\alpha_1 = \ln\frac{k_1}{k_2} + k_1 \ln x_{m_1} - k_2 \ln x_{m_2}, \quad \beta_1 = k_2 - k_1, \quad h(x) = \ln x.$$

11. Distribusi Bernoulli.

$$\frac{g_1(x)}{g_2(x)} = \frac{p_1^x (1-p_1)^{1-x}}{p_2^x (1-p_2)^{1-x}}$$

$$= \left(\frac{1-p_1}{1-p_2}\right)\left(\frac{p_1}{p_2}\right)^x \left(\frac{1-p_1}{1-p_2}\right)^{-x}$$

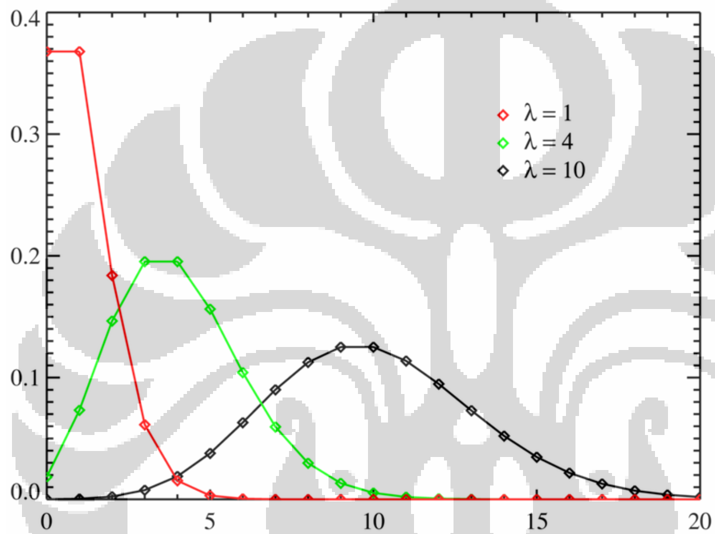
$$= \exp\left\{ \log\frac{1-p_1}{1-p_2} + x\log\frac{p_1}{p_2} - x\log\frac{1-p_1}{1-p_2} \right\}$$

$$= \exp\left\{ \log\frac{1-p_1}{1-p_2} + \left(\log\frac{p_1}{p_2} - \log\frac{1-p_1}{1-p_2} \right).x \right\}$$

sehingga diperoleh

$$\alpha_1 = \log\frac{1-p_1}{1-p_2}, \quad \beta_1 = \log\frac{p_1}{p_2} - \log\frac{1-p_1}{1-p_2}, \quad h(x) = x$$

12. Distribusi Poisson.



$$\frac{g_1(x)}{g_2(x)} = \frac{\dfrac{\lambda_1^x e^{-\lambda_1}}{x!}}{\dfrac{\lambda_2^x e^{-\lambda_2}}{x!}}$$

$$= \frac{e^{-\lambda_1}}{e^{-\lambda_2}}\left(\frac{\lambda_1}{\lambda_2}\right)^x$$

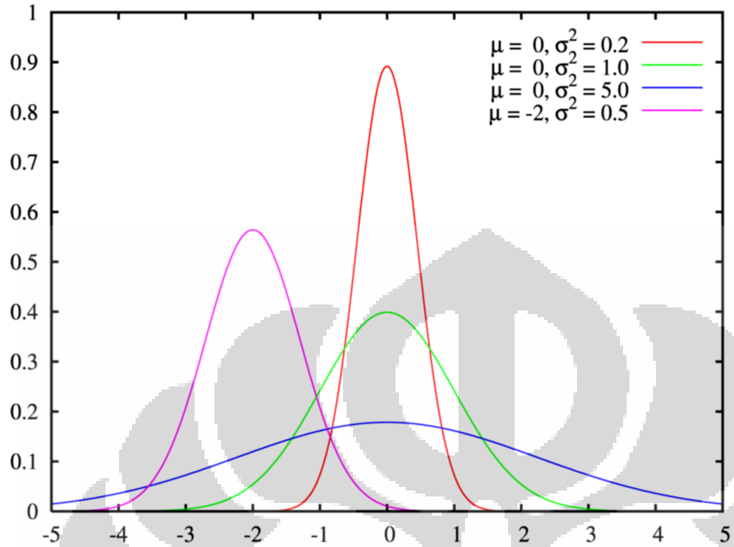
$$= \exp\left\{-\lambda_1 + \lambda_2 + \left(\log\frac{\lambda_1}{\lambda_2}\right)x\right\}$$

$$= \exp\left\{-(\lambda_1 - \lambda_2) + \left(\log\frac{\lambda_1}{\lambda_2}\right)x\right\}$$

sehingga diperoleh

$$\alpha_1 = -(\lambda_1 - \lambda_2), \quad \beta_1 = \log\frac{\lambda_1}{\lambda_2}, \quad h(x) = x$$

13. Distribusi Normal.



Untuk $N(\mu,\sigma^2)$, dengan mean dan variansi yang tidak sama,

$$\frac{g_1(x)}{g_2(x)} = \frac{\dfrac{1}{\sigma_1\sqrt{2\pi}}\exp\left\{-\dfrac{(x-\mu_1)^2}{2\sigma_1^2}\right\}}{\dfrac{1}{\sigma_2\sqrt{2\pi}}\exp\left\{-\dfrac{(x-\mu_2)^2}{2\sigma_2^2}\right\}}$$

$$= \frac{\sigma_2 \exp\left\{-\dfrac{(x-\mu_1)^2}{2\sigma_1^2}\right\}}{\sigma_1 \exp\left\{-\dfrac{(x-\mu_2)^2}{2\sigma_2^2}\right\}}$$

$$= \exp\left\{\ln\left(\frac{\sigma_2}{\sigma_1}\right) - \frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{(x-\mu_2)^2}{2\sigma_2^2}\right\}$$

$$= \exp\left\{\ln\left(\frac{\sigma_2}{\sigma_1}\right) - \frac{(x^2-2\mu_1 x+\mu_1^2)}{2\sigma_1^2} + \frac{(x^2-2\mu_2 x+\mu_2^2)}{2\sigma_2^2}\right\}$$

$$= \exp\left\{\ln\left(\frac{\sigma_2}{\sigma_1}\right) - \frac{x^2}{2\sigma_1^2} + \frac{\mu_1 x}{\sigma_1^2} - \frac{\mu_1^2}{2\sigma_1^2} + \frac{x^2}{2\sigma_2^2} - \frac{\mu_2 x}{\sigma_2^2} + \frac{\mu_2^2}{2\sigma_2^2}\right\}$$

$$= \exp\left\{\ln\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\mu_2^2}{2\sigma_2^2} - \frac{\mu_1^2}{2\sigma_1^2} + \left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right)x + \left(\frac{1}{2\sigma_2^2} - \frac{1}{2\sigma_1^2}\right)x^2\right\}$$

$$= \exp\left\{ \ln\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\mu_2^2}{2\sigma_2^2} - \frac{\mu_1^2}{2\sigma_1^2} + \left(\frac{\mu_1\sigma_2^2 - \mu_2\sigma_1^2}{\sigma_1^2\sigma_2^2}\right)x + \left(\frac{\sigma_1^2 - \sigma_2^2}{2\sigma_1^2\sigma_2^2}\right)x^2 \right\}$$

$$= \exp\left\{ \left(\ln\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\mu_2^2}{2\sigma_2^2} - \frac{\mu_1^2}{2\sigma_1^2} \right) + \begin{pmatrix} (\mu_1\sigma_2^2 - \mu_2\sigma_1^2)/\sigma_1^2\sigma_2^2 \\ (\sigma_1^2 - \sigma_2^2)/2\sigma_1^2\sigma_2^2 \end{pmatrix} \quad x^2 \right\}$$

sehingga diperoleh

$$\alpha_1 = \ln\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\mu_2^2}{2\sigma_2^2} - \frac{\mu_1^2}{2\sigma_1^2}$$

$$\boldsymbol{\beta}_1 = \begin{pmatrix} \beta_{11} \\ \beta_{12} \end{pmatrix} = \begin{pmatrix} (\mu_1\sigma_2^2 - \mu_2\sigma_1^2)/\sigma_1^2\sigma_2^2 \\ (\sigma_1^2 - \sigma_2^2)/2\sigma_1^2\sigma_2^2 \end{pmatrix}$$

$\boldsymbol{h}(x) = (x, x^2)'$

Lampiran 4

Metode Newton – Raphson

Metode Newton – Raphson merupakan suatu metode (iterasi) numerik yang paling *powerful* untuk mencari solusi akar suatu persamaan.

Definisikan:

$C[a,b]$ adalah himpunan semua fungsi kontinu pada interval tertutup [*a*, *b*].

$C^n(X)$ adalah himpunan semua fungsi yang memiliki *n* turunan yang kontinu pada *X*.

Misalkan $f \in C^2[a,b]$, $\bar{x} \in [a,b]$ merupakan aproksimasi dari *p* sedemikian sehingga $f'(\bar{x}) \neq 0$ dan $|p-\bar{x}|$ 'kecil'. Polinomial Taylor untuk *f*(*x*) di sekitar $\bar{x}$ adalah

$$f(x) = f(\bar{x}) + (x-\bar{x})f'(\bar{x}) + \frac{(x-\bar{x})^2}{2} + f''(\xi(x))$$

dimana $\xi(x)$ terletak berada di antara *x* dan $\bar{x}$. Dikarenakan $f(p)=0$, persamaan di atas dengan $x=p$ adalah

$$0 = f(\bar{x}) + (p-\bar{x})f'(\bar{x}) + \frac{(p-\bar{x})^2}{2} + f''(\xi(p))$$

Pada metode Newton – Raphson diasumsikan $|p-\bar{x}|$ 'kecil', suku $p-\bar{x}\ ^2$ memiliki nilai jauh lebih kecil, sehingga

$$0 \approx f(\bar{x}) + (p - \bar{x}) f'(\bar{x})$$

Solusi untuk *p*

$$p \approx \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}$$

Tahap pada metode Newton – Raphson, dimulai dengan aproksimasi awal

p_0 dan membangun barisan $p_n \, {}_{n=0}^{\infty}$, oleh

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

Dengan toleransi $\varepsilon > 0$, Iterasi ini akan berhenti hingga terpenuhi

$$|p_N - p_{N-1}| < \varepsilon$$

Metode Newton – Raphson merupakan teknik iterasi fungsional dari bentuk

$p_n = g(n-1)$, dimana

$$g(n-1) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$, untuk $n \geq 1$.

Teknik iterasi fungsional tersebut memberikan konvergensi yang cepat.

Lampiran 5

Program S-PLUS

```
# Functions for Kedem density ratio semi-parametric model for comparing
# different distributions, for instance, ANOVA
# minusloglike=-logL, because later we use minimize -logL,
# which means maximize the L
# theta is the parameter vector need to be estimated
minusloglike <- function(theta,datasize,t){
       # parameters need to be calculated
       n<-sum(datasize)
       m<-length(datasize)
       rho<-datasize[1:(m-1)]/datasize[m]
       comboh<-matrix(c(rep(1,n),h(t)),length(c(rep(1,n),h(t)))/n,n,byrow=T)

       thetamax<-matrix(theta[1:((m-1)*dim(comboh)[1])],m-
    1,dim(comboh)[1],byrow=T)
       allelements<-thetamax%*%comboh
       logterm<-sum(log(1+rho%*%exp(allelements)))

       ind<-cumsum(datasize)
       resterm<-0
       resterm<-sum(allelements[1,1:ind[1]])
       q<-2
       while (q<=m-1) {
          resterm<-resterm+sum(allelements[q,(1+ind[q-1]):ind[q]])
          q<-q+1
       }
       minusloglike<-logterm-resterm
   }
# the gradiant of the -loglikelihood function
musL.g <- function(theta,datasize,t){
       # parameters need to be calculated
       n<-sum(datasize)
       m<-length(datasize)
       rho<-datasize[1:(m-1)]/datasize[m]
       comboh<-matrix(c(rep(1,n),h(t)),length(c(rep(1,n),h(t)))/n,n,byrow=T)

       thetamax<-matrix(theta[1:((m-1)*dim(comboh)[1])],m-
    1,dim(comboh)[1],byrow=T)
       allelementes<-exp(thetamax%*%comboh)
       fenzie<-rho*allelementes

       firsterm<-(kronecker(fenzie, rep(1,dim(comboh)[1]), fun="*")*
                  kronecker(rep(1,m-1), comboh, fun="*"))%*%
                  t(1/(1+rho%*%allelementes))

       ind<-cumsum(datasize)
       secondterm<-rowSums(comboh[,1:datasize[1]])
       q<-2
       while (q<=m-1) {
          secondterm<-c(secondterm, rowSums(comboh[,(1+ind[q-1]):ind[q]]))
          q<-q+1
       }
       # should be a q(pp+1)*1 vector
       musL.g<- firsterm - secondterm
```

```
}


# Function: Kedem density ratio for comparing different distributions, for
   instance, ANOVA
SemiCombw<-
   function(datasize,t,h,initials,H,C,plotinc,testdirc,chi2compute,messagesh
   ow){
   # Arguements:
   #   datasize: sample size for each sample
   #   h: the h function, h(), to show what kind of term I should include,
   #      Some Choices of h(x) function
   #        h <- function(x){x}      for traditional ANOVA. Good to start
   with.
   #        h <- function(x){log(x)} for skewed data, it is bot bad.
   #        h <- function(x){x,x^2}  for test ANOVA when variances are
   different.
   #   initials: the initial values of theta, which is the parameter to be
   estimated
   #   H: H matrix used in Wald type Chi2 test
   #   C: C vector used in Wald type Chi2 test
   #   plotinc: Plot increment, if 0 don't plot
   #
   # Please also see the example below for better understanding
   #
   # h<-function(x){eval(hx)}  (no use)


   # common parameters need to be calculated
   n<-sum(datasize)
   m<-length(datasize)
   q<-m-1
   rho<-datasize[1:q]/datasize[m]
   comboh<-matrix(c(rep(1,n),h(t)),length(c(rep(1,n),h(t)))/n,n,byrow=T)
   pp<-dim(comboh)[1]-1

    #Parameter estimates
    if (initials[1]=="D") initials<-0.02*runif(q*(1+pp),-1,1)
   minfunc<-nlminb(start=initials,obj = minusloglike,
                   gradient=musL.g,
                   datasize=datasize, t=t)    # maximize likelihood
   thetahead<-minfunc$parameters
   numitera<-minfunc$iterations
   thetahdmax<-matrix(thetahead,q,dim(comboh)[1],byrow=T)

   thetahdalpha<-thetahdmax[,1]
   thetahdbeta<-t(thetahdmax[,2:(pp+1)])
   # thetahdvec in the order of alpha1~q, beta1_1~pp, ..., betaq_1~pp
   thetahdvec<-matrix(thetahdalpha,q,1)
   thetahdvec<-rbind(thetahdvec,matrix(thetahdbeta,q*pp,1))

   # Estimated unsmoothed density of reference distribution, g and distorted
   ones.
   w<-exp(thetahdmax%*%comboh)            # wj(t)   q by n
   p<-c(1/(datasize[m]*(1+matrix(rho,1,q)%*%w)))
   distortedp<-matrix(0,n,q)
   for (j in 1:q)
      distortedp[,j]<-matrix(w[j,]*p,n,1)
   allp<-cbind(distortedp,matrix(p,n,1))
```

```
allmean<-rep(0,m)
allvar<-rep(0,m)
for (j in 1:m){
   allmean[j]<-sum(t*c(allp[,j]))
    allvar[j]<-sum(t^2*c(allp[,j]))-allmean[j]^2
}



# Calculate the V and S for matrix S^-1VS^-1
w<-rbind(w,rep(1,n))           #  wj(t)  m by n

## Estimate of Ej(t)=Ej[h(t)] (varj(t)=Varj[h(t)]) w.r.t. reference dist.
g and distorted g.
Eht<-matrix(0,pp,m)
for (j in 1:m){
  for (i in 1:pp)
      Eht[i,j]<-sum(comboh[i+1,]*w[j,]*p)   # coincidently the same as
mean(xm)
}

## Calculate A matrices
Denomina<-matrix(c(rho,1),1,m)%*%w
A0<-matrix(0,m,m)
A1<-matrix(0,pp*m,m)
A2<-matrix(0,pp*m,pp*m)
for (j in 1:m){
   for (jj in 1:m) {
      A0[j,jj]<-sum(w[j,]*w[jj,]*p/Denomina)
      for (i in 1:pp){
         A1[(j-1)*pp+i,jj]<-sum(comboh[i+1,]*w[j,]*w[jj,]*p/Denomina)
         for (k in 1:pp)
             A2[(j-1)*pp+i,(jj-1)*pp+k]<-
sum(comboh[i+1,]*comboh[k+1,]*w[j,]*w[jj,]*p/Denomina)
      }
   }
}

## Calculate the V matrix
Valpha<-matrix(0,q,q)
for (j in 1:q){
  for (jj in 1:q){
      if (j==jj)
         Valpha[j,jj]<-rho[j]^2*(A0[j,j]-
sum(c(rho,1)*c(A0[j,])^2))/(1+sum(rho))
      else
         Valpha[j,jj]<-rho[j]*rho[jj]*(A0[j,jj]-
sum(c(rho,1)*c(A0[j,])*c(A0[jj,])))/(1+sum(rho))
  }
}

Vabup<-matrix(0,q,pp*q)
for (j in 1:q){
   for (jj in 1:q){
      if (j==jj){
         for (i in 1:pp)
            Vabup[j,(jj-1)*pp+i]<-rho[j]^2*(A0[j,j]*Eht[i,j]-
sum(c(rho,1)*c(A0[j,])*A1[i+pp*c(0,seq(q)),j]))/(1+sum(rho))
      }
      else {
```

```
          for (i in 1:pp)
            Vabup[j,(jj-1)*pp+i]<-rho[j]*rho[jj]*(A0[j,jj]*Eht[i,jj]-
sum(c(rho,1)*c(A0[j,])*A1[i+pp*c(0,seq(q)),jj]))/(1+sum(rho))

      }
    }
 }
 Vabdown<-t(Vabup)

 Vbeta<-matrix(0,pp*q,pp*q)
 for (j in 1:q){
   for (jj in 1:q){
      subterm<-0
       for (r in 1:q)
         subterm<-subterm+rho[r]*A1[(j-1)*pp+seq(pp),r]%*%t(A1[(jj-
1)*pp+seq(pp),r])
       subterm<-subterm+1*A1[(j-1)*pp+seq(pp),m]%*%t(A1[(jj-
1)*pp+seq(pp),m])

      if (j==jj){
         Vht<-matrix(0,pp,pp)
         for (i in 1:pp) {
             for (k in 1:pp)
                Vht[i,k]<-sum(comboh[i+1,]*comboh[k+1,]*w[j,]*p)-
Eht[i,j]*Eht[k,j]
          }
          Vbeta[(j-1)*pp+seq(pp),(jj-1)*pp+seq(pp)]<-rho[j]*rho[jj]*(-
A2[(j-1)*pp+seq(pp),(jj-1)*pp+seq(pp)]+Eht[,j]%*%t(A1[(j-
1)*pp+seq(pp),jj])+A1[(j-1)*pp+seq(pp),jj]%*%t(Eht[,jj])-
subterm)/(1+sum(rho))+(datasize[j]/n)*Vht
      }
      else
         Vbeta[(j-1)*pp+seq(pp),(jj-1)*pp+seq(pp)]<-rho[j]*rho[jj]*(-
A2[(j-1)*pp+seq(pp),(jj-1)*pp+seq(pp)]+Eht[,j]%*%t(A1[(j-
1)*pp+seq(pp),jj])+A1[(j-1)*pp+seq(pp),jj]%*%t(Eht[,jj])-
subterm)/(1+sum(rho))
    }
  }

 for (i in 1:pp) {
    for (k in 1:pp)
        Vht[i,k]<-sum(comboh[i+1,]*comboh[k+1,]*w[m,]*p)-Eht[i,m]*Eht[k,m]
 }

V1<-cbind(Valpha,Vabup)
V2<-cbind(Vabdown,Vbeta)
V<-rbind(V1,V2)


# A11 matrix of the S matrix
A11<-matrix(0,q,q)
for (j in 1:q){
  for (jj in 1:q){
      if (j==jj)
         A11[j,jj]<-rho[j]*(1+sum(rho)-rho[j])/(1+sum(rho))^2
      else
         A11[j,jj]<--rho[j]*rho[jj]/(1+sum(rho))^2
   }
}
```

```
#The Chi1 Test Statistic:
sigmainv<-kronecker(A11,Vht, fun="*")
Chi1 <-
 n*t(thetahdvec[seq(q+1,(pp+1)*q),1])%*%sigmainv%*%(thetahdvec[(q+1):((pp+
 1)*q),1])
pvalue.Chi1<- 1-pchisq(Chi1,pp*q)


# calculate Chi2 if  chi2compute=='Y'
if (chi2compute=='Y') {

## Calculate the S matrix
Salpha<-matrix(0,q,q)
for (j in 1:q){
  for (jj in 1:q){
      if (j==jj)
         Salpha[j,jj]<-rho[j]*sum((Denomina-
 rho[j]*w[j,])*w[j,]*p/Denomina)/(1+sum(rho))
      else
         Salpha[j,jj]<--
 rho[j]*rho[jj]*sum(w[j,]*w[jj,]*p/Denomina)/(1+sum(rho))
   }
}

Sabup<-matrix(0,q,pp*q)
for (j in 1:q){
   for (jj in 1:q){
      if (j==jj){
         for (i in 1:pp)
            Sabup[j,(jj-1)*pp+i]<-rho[j]*sum((Denomina-
 rho[j]*w[j,])*w[j,]*comboh[i+1,]*p/Denomina)/(1+sum(rho))
      }
      else {
          for (i in 1:pp)
             Sabup[j,(jj-1)*pp+i]<--
 rho[j]*rho[jj]*sum(w[j,]*w[jj,]*comboh[i+1,]*p/Denomina)/(1+sum(rho))
      }
    }
}
Sabdown<-t(Sabup)

Sbeta<-matrix(0,pp*q,pp*q)
for (j in 1:q){
  for (jj in 1:q){
     if (j==jj){
         for (i in 1:pp){
           for (k in 1:pp)
               Sbeta[(j-1)*pp+i,(jj-1)*pp+k]<-rho[j]*sum((Denomina-
 rho[j]*w[j,])*w[j,]*comboh[i+1,]*comboh[k+1,]*p/Denomina)/(1+sum(rho))
         }
     }
     else{
         for (i in 1:pp){
           for (k in 1:pp)
               Sbeta[(j-1)*pp+i,(jj-1)*pp+k]<--
 rho[j]*rho[jj]*sum(w[j,]*w[jj,]*comboh[i+1,]*comboh[k+1,]*p/Denomina)/(1+
 sum(rho))
          }
     }
    }
```

```
}

S1<-cbind(Salpha,Sabup)
S2<-cbind(Sabdown,Sbeta)
S<-rbind(S1,S2)
accuS<-'True'

#cat('Sinverse=',S,fill=T)

## Calculate Sigma
if (qr(S)$rank<dim(S)[1]){
    if (messageshow==1)
       cat("S is Singular, so Sigma is not real one, only trust Chi1 test
or Likelihood ratio test", fill=T)
    accuS<-'No'
    Chi2 <- -1
    pvalue.Chi2<- -1
    Sigma<-solve(sigmainv)
}
else {
   Sigma<-solve(S)%*%V%*%solve(S)

   ## Standard Errors can be obtained by square root of Sigma
   paraSE<-diag(sqrt(Sigma))/sqrt(n)    # in the order of alpha1~q,
beta1_1~pp, ..., betaq_1~pp

   #The Chi2 Test Statistic: (Sometimes S^-1HS^-1 is singular)
   if (H[1]=="D") H<-cbind(matrix(0,q*pp,q),diag(rep(1,q*pp)))
   if (C[1]=="D") C<-rep(0,dim(H)[1])
   Chi2 <- n*t(H%*%thetahdvec-
C)%*%solve(H%*%Sigma%*%t(H))%*%(H%*%thetahdvec-C)
   pvalue.Chi2<- 1-pchisq(Chi2,dim(H)[1])
 }
}
else {
    Sigma<-solve(sigmainv)
    if (messageshow==1)
         cat("Chi2 not computed, Standard error from Chi1.", fill=T)
   accuS<-'No'
   Chi2 <- -1
   pvalue.Chi2<- -1
   S<-'NA'
}


## Standard Errors can be obtained by square root of Sigma
paraSE<-diag(sqrt(Sigma))/sqrt(n)    # in the order of alpha1~q,
beta1_1~pp, ..., betaq_1~pp


#Likelihood ratio (-2Log(^)) Test Statistic:
allele<-matrix(w[1:q,],q,n,byrow=F)  # equivalent to   log(w[1:q,]) when
q=1
inde<-cumsum(datasize)
abterm<-sum(log(allele[1,1:inde[1]]))
qi<-2
while (qi<=q) {
   abterm<-abterm+sum(log(allele[qi,(1+inde[qi-1]):inde[qi]]))
   qi<-qi+1
}
```

```
LogLRts<-2*n*log(1+sum(rho))-
 2*sum(log(matrix(c(rho,1),1,m)%*%w))+2*abterm
pvalue.LogLRts<- 1-pchisq(LogLRts,pp*q)


# only use Chi2 test for 1-sided test
# Chi2 <- sqrt(n)%*%(H%*%thetahdvec-C)%*%sqrt(solve(H%*%Sigma%*%t(H)))
 # pvalue.Chi2<-ifelse(Chi2>0, 1-pnorm(Chi2), pnorm(Chi2))

if (testdirc==">"){         # hot
   ts1sd<-sqrt(Chi1)*sign(thetahdbeta)
   pvalue1sd<-1-pnorm(ts1sd)
}
else if(testdirc=="<"){      # cold
    ts1sd<-sqrt(Chi1)*sign(thetahdbeta)
    pvalue1sd<-pnorm(ts1sd)
}
else{                    # two sided
   ts1sd<-'NA'
   pvalue1sd<-'NA'
}

# Monte Carlo hypothesis testing method
# need to sample from the estimated distribution, which is complicated
 !!!


# ## As Kedem suggested, maybe a simplified one ####
#
# ### Estimate of Ej(t)=Ej[h(t)], and var[h(t)]all the same actually
# Eht<-comboh[2:(pp+1),]%*%matrix(p,n,1)
# Eht2<-matrix(0,pp,pp)
# Vht<-matrix(0,pp,pp)
# for (i in 1:pp) {
#    for (k in 1:pp){
#       Eht2[i,k]<-sum(comboh[i+1,]*comboh[k+1,]*p)
 #      Vht[i,k]<-sum(comboh[i+1,]*comboh[k+1,]*p)-Eht[i]*Eht[k]
 #   }
 # }
#
# ### Calculate the S matrix
#  S1<-cbind(A11,kronecker(A11, t(Eht), fun="*"))
# S2<-cbind(kronecker(A11, Eht, fun="*"),kronecker(A11, Eht2, fun="*"))
# S<-rbind(S1,S2)
# ### Calculate the V matrix
# V<-rbind(matrix(0,q,(pp+1)*q),cbind(matrix(0,pp*q,q),kronecker(A11,
 Vht, fun="*")))
# ### Calculate Sigma
# SigmaSim<-solve(S)%*%V%*%solve(S)
# ### The simplified Chi1 Test Statistic:
# Chi1old <-
 n*t(thetahdvec[seq(q+1,(pp+1)*q),1])%*%solve(SigmaSim[seq(q+1,(pp+1)*q),s
 eq(q+1,(pp+1)*q)])%*%(thetahdvec[(q+1):((pp+1)*q),1])
# pvalue.Chi1old<- 1-pchisq(Chi1old,pp*q)
#
# ### Simplified one END #####

 # Prepare for plot
if (plotinc>0)
   Increment <-plotinc
```

```
else
   Increment <-(max(t)-min(t))/100

x <- seq(min(t),max(t),Increment)
cdf <-rep(0,length(x))
for(i in 1:length(x)) cdf[i]<-sum(p[t <= x[i]])
allcdf<-matrix(0,length(x),q)
for (j in 1:q) {
    distortedpj<-distortedp[,j]
    for(i in 1:length(x)) allcdf[i,j]<-sum(distortedpj[t <= x[i]])
}
allcdf<-cbind(allcdf,cdf)      # CDF

# Estimated Unsmoothed density g
allg<-rbind(0,diff(allcdf)/Increment)   # diff(x) should be diff(x)
g<-c(0,diff(cdf)/Increment)

if (plotinc>0){
   newcomboh<-
matrix(c(rep(1,length(x)),h(x)),length(c(rep(1,length(x)),h(x)))/length(x
),length(x),byrow=T)
   temp<-exp(thetahdmax%*%newcomboh)

   # plot Reference CDF G
   plot(x,allcdf[,m], type="l", xlab="x", ylab="Estimated G(t)",col=1,
        lty=1,xlim=c(min(t),max(t)),ylim=c(min(allcdf),max(allcdf)))
   # plot distorted CDF
   for (j in 1:q) lines(x,allcdf[,j], type="l",col=(j+1),lty=(j+1))

   # add legend for the CDF plot
   legend((min(t)+diff(range(t))/25),1.0,
          legend=c(paste("sample",1:q),"sample ref"),
          lty=c(2:m,1), col=c(2:m,1))

   #Smoothed g
   plot(smooth.spline(x,allg[,m]), type="l", xlab="x", ylab="Est.
Density",col=1,
        lty=1,xlim=c(min(t),max(t)),ylim=c(min(allg),max(allg)))
   abline(v=allmean[m],col=1)

   #Distorted pdf for h(x)=hx
   for (j in 1:q) {
      lines(smooth.spline(x,c(temp[j,])*g),type="l",col=(j+1),lty=(j+1))
      abline(v=allmean[j],col=(j+1))
      #abline(v=sum(x*allg[,j]*Increment),col=j)
   }

   # add legend for the pdf plot
   legend((max(t)-diff(range(t))/3),(max(allg)-diff(range(allg))/20),
          legend=c(paste("sample",1:q),"sample ref"),
          lty=c(2:m,1), col=c(2:m,1))


}

# Watch window, only for debugging
# SemiCombw<-1+sum(rho)-rho[1]
# ww<-c(Chi1old,pvalue.Chi1old)

# Output result
```

```
    list(alpha=thetahdalpha,beta=thetahdbeta,alphaSE=paraSE[1:q],betaSE=matri
    x(paraSE[(q+1):length(paraSE)],pp,q),
         dataEstmean=allmean,dataEstvar=allvar, accurateS=accuS,
         Chi2=Chi2,Pvalue2=pvalue.Chi2,Sigma2=Sigma,
         Chi1=Chi1,Pvalue1=pvalue.Chi1, zts1sd=ts1sd, p1sd=pvalue1sd,
    watch=abterm,
         Chi3=LogLRts,Pvalue3=pvalue.LogLRts, initialsUsed=initials,
         cdf=allcdf,unsmothden=allg,stepval=x,refp=p,ni=numitera,
         ml=-minfunc$objective, Smatrix=S, Vmatrix=V)
   # add ",watch=ww" into the list function iff debugging
   # Sigma1=SigmaSim is abandoned since not good performance
}

########################## SemiKedem end #############################


############# An example to show how to use this function #############

 x1 <- c(758, 20374,2184,10924,25851)
 x2 <- c(24932,2306,1402,58,2272,129,365,969,1067,948,2125)

 sizsofsamples<-c(length(x1),length(x2))
 combndata<-c(x1,x2)
 h<-function(x){c(x,1/x)}   # pp= number of terms in the c(..) of function h
 inivalues<-"D"          #c(-0.01,0,0,0,0.01,0,0,0,0.02),  length:  q*(1+pp)
 plotincremt<-(max(combndata)-min(combndata))/100 #if =0 or-1,no plot output
 res<-SemiCombw(sizsofsamples,combndata,h,inivalues,Hmatrix,Cvector,
   plotincremt,"!=",'Y',1)
 res$alpha
 res$beta
 res$Chi3
 res$Chi2
 c(res$Pvalue1, res$Pvalue2, res$Pvalue3)
 round(res$initialsUsed,digit=4)
```