Estimation of Dynamic Latent Variable Models Using Simulated Nonparametric Moments

Michael Creel

Universitat Autònoma de Barcelona

UFAE Macro Workshop, March 2008

Outline

The DLV model

Billio and Monfort (2003)

$$\text{DLV:} \begin{cases} y_t = & r_t\left(y^{t-1}, y^{*t}, \varepsilon_t; \theta\right) \\ y_t^* = & r_t^*\left(y^{t-1}, y^{*t-1}, \varepsilon_t^*; \theta\right) \end{cases} \tag{1}$$

- y^{t-1} is notation for $\left(y_1', ..., y_{t-1}'\right)'$
- $\{\varepsilon_t\}$ and $\{\varepsilon_t^*\}$ are two independent white noises with known distributions
- θ is a vector of unknown parameters

The DLV model

Billio and Monfort (2003)

$$\text{DLV:} \begin{cases} y_t = & r_t\left(y^{t-1}, y^{*t}, \varepsilon_t; \theta\right) \\ y_t^* = & r_t^*\left(y^{t-1}, y^{*t-1}, \varepsilon_t^*; \theta\right) \end{cases} \quad (1)$$

- y^{t-1} is notation for $\left(y_1', ..., y_{t-1}'\right)'$
- $\{\varepsilon_t\}$ and $\{\varepsilon_t^*\}$ are two independent white noises with known distributions
- θ is a vector of unknown parameters

The DLV model

Billio and Monfort (2003)

$$\text{DLV:}\quad \begin{cases} y_t = & r_t\left(y^{t-1}, y^{*t}, \varepsilon_t; \theta\right) \\ y_t^* = & r_t^*\left(y^{t-1}, y^{*t-1}, \varepsilon_t^*; \theta\right) \end{cases} \tag{1}$$

- y^{t-1} is notation for $\left(y_1', \ldots, y_{t-1}'\right)'$
- $\{\varepsilon_t\}$ and $\{\varepsilon_t^*\}$ are two independent white noises with known distributions
- θ is a vector of unknown parameters

The DLV model

Billio and Monfort (2003)

$$\text{DLV:} \begin{cases} y_t = & r_t\left(y^{t-1}, y^{*t}, \varepsilon_t; \theta\right) \\ y_t^* = & r_t^*\left(y^{t-1}, y^{*t-1}, \varepsilon_t^*; \theta\right) \end{cases} \quad (1)$$

- y^{t-1} is notation for $\left(y_1', ..., y_{t-1}'\right)'$
- $\{\varepsilon_t\}$ and $\{\varepsilon_t^*\}$ are two independent white noises with known distributions
- θ is a vector of unknown parameters

DLV models are often impossible to estimate using classical methods

- Calculation of the likelihood function requires finding the density of y^n
- this involves integrating out all of the y_t^*, of which there are n. It is in general an untractable problem
- Without the density of the observable variables, analytic moments cannot be computed
- Without the density function, maximum likelihood is unavailable
- Without moments, moment-based estimation methods are not available

DLV models are often impossible to estimate using classical methods

- Calculation of the likelihood function requires finding the density of y^n
- this involves integrating out all of the y_t^*, of which there are n. It is in general an untractable problem
- Without the density of the observable variables, analytic moments cannot be computed
- Without the density function, maximum likelihood is unavailable
- Without moments, moment-based estimation methods are not available

DLV models are often impossible to estimate using classical methods

- Calculation of the likelihood function requires finding the density of y^n
- this involves integrating out all of the y_t^*, of which there are n. It is in general an untractable problem
- Without the density of the observable variables, analytic moments cannot be computed
- Without the density function, maximum likelihood is unavailable
- Without moments, moment-based estimation methods are not available

DLV models are often impossible to estimate using classical methods

- Calculation of the likelihood function requires finding the density of y^n
- this involves integrating out all of the y_t^*, of which there are n. It is in general an untractable problem
- Without the density of the observable variables, analytic moments cannot be computed
- Without the density function, maximum likelihood is unavailable
- Without moments, moment-based estimation methods are not available

DLV models are often impossible to estimate using classical methods

- Calculation of the likelihood function requires finding the density of y^n
- this involves integrating out all of the y_t^*, of which there are n. It is in general an untractable problem
- Without the density of the observable variables, analytic moments cannot be computed
- Without the density function, maximum likelihood is unavailable
- Without moments, moment-based estimation methods are not available

Some of the existing simulation-based estimation methods

- simulated method of moments (McFadden, 1989; Pakes and Pollard, 1989)
- indirect inference (Gouriéroux, Monfort and Renault, 1993; Smith, 1993)
- simulated pseudo-maximum likelihood (Laroque and Salanié, 1993)
- simulated maximum likelihood (Lee, 1995)

... more methods

- efficient method of moments (Gallant and Tauchen, 1996)
- method of simulated scores (Hajivassiliou and McFadden, 1998)
- kernel-based indirect inference (Billio and Monfort, 2003)
- simulated EM algorithm (Fiorentini, Sentana and Shephard, 2004)
- nonparametric simulated maximum likelihood (Fermanian and Salanié, 2004; Kristensen and Shin, 2006)
- simulated nonparametric estimators (Altissimo and Mele, 2007)

Efficiency and Generality don't go hand in hand

General purpose

- SMM
- indirect inference
- simulated pseudo-maximum likelihood
- EMM
- KBII
- NPSML
- SNE

Efficient for DLV models

- EMM

Efficiency and Generality don't go hand in hand

General purpose

- SMM
- indirect inference
- simulated pseudo-maximum likelihood
- EMM
- KBII
- NPSML
- SNE

Efficient for DLV models

- EMM

Efficiency and Generality don't go hand in hand

General purpose

- SMM
- indirect inference
- simulated pseudo-maximum likelihood
- EMM
- KBII
- NPSML
- SNE

Efficient for DLV models

- EMM

Efficiency and Generality don't go hand in hand

General purpose

- SMM
- indirect inference
- simulated pseudo-maximum likelihood
- EMM
- KBII
- NPSML
- SNE

Efficient for DLV models

- EMM

Why is SMM inefficient?

- SMM is inefficient for DLV models, since conditional moments can't be used. Why not?
- It's because we can't sample the latent variables conditional on the history of the observed variables. Recall the model:

$$\text{DLV: } \left\{ y_t = \quad r_t\left(y^{t-1}, y^{*t}, \varepsilon_t; \theta\right) \right. \tag{2}$$

  If we could sample from $y^{*t}|y^{t-1}$, we could substitute the draw into the DLV to get a draw from $y_t|y^{t-1}$
- For Markovian models, the Markov chain Monte Carlo method can be used to sample from $y^{*t}|y^{t-1}$, since in this case information about the distant past is not needed to sample the latent variables. Fiorentini, Sentana and Shephard (2004) is an example.

Why is SMM inefficient?

- SMM is inefficient for DLV models, since conditional moments can't be used. Why not?
- It's because we can't sample the latent variables conditional on the history of the observed variables. Recall the model:

$$\text{DLV:} \left\{ y_t = \quad r_t \left(y^{t-1}, y^{*t}, \varepsilon_t; \theta \right) \right. \tag{2}$$

  If we could sample from $y^{*t}|y^{t-1}$, we could substitute the draw into the DLV to get a draw from $y_t|y^{t-1}$

- For Markovian models, the Markov chain Monte Carlo method can be used to sample from $y^{*t}|y^{t-1}$, since in this case information about the distant past is not needed to sample the latent variables. Fiorentini, Sentana and Shephard (2004) is an example.

Why is SMM inefficient?

- SMM is inefficient for DLV models, since conditional moments can't be used. Why not?
- It's because we can't sample the latent variables conditional on the history of the observed variables. Recall the model:

$$\text{DLV: } \left\{ y_t = \quad r_t\left(y^{t-1}, y^{*t}, \varepsilon_t; \theta\right) \right. \tag{2}$$

  If we could sample from $y^{*t}|y^{t-1}$, we could substitute the draw into the DLV to get a draw from $y_t|y^{t-1}$
- For Markovian models, the Markov chain Monte Carlo method can be used to sample from $y^{*t}|y^{t-1}$, since in this case information about the distant past is not needed to sample the latent variables. Fiorentini, Sentana and Shephard (2004) is an example.

The GMM estimator

- Error functions are of the form

$$\varepsilon(y_t, x_t; \theta) = y_t - \phi(x_t; \theta), \tag{3}$$

- Moment conditions are defined by interacting a vector of instrumental variables $z(x_t)$ with error functions:

$$m(y_t, x_t; \theta) = z(x_t) \otimes \varepsilon(y_t, x_t; \theta) \tag{4}$$

- Average moment conditions are

$$m_n(Z_n; \theta) = \frac{1}{n} \sum_{t=1}^{n} m(y_t, x_t; \theta) \tag{5}$$

- The objective function is

$$s_n(Z_n; \theta) = m_n'(Z_n; \theta) W(\hat{\tau}_n) m_n'(Z_n; \theta) \tag{6}$$

The GMM estimator

- Error functions are of the form

$$\varepsilon(y_t, x_t; \theta) = y_t - \phi(x_t; \theta), \tag{3}$$

- Moment conditions are defined by interacting a vector of instrumental variables $z(x_t)$ with error functions:

$$m(y_t, x_t; \theta) = z(x_t) \otimes \varepsilon(y_t, x_t; \theta) \tag{4}$$

- Average moment conditions are

$$m_n(Z_n; \theta) = \frac{1}{n} \sum_{t=1}^{n} m(y_t, x_t; \theta) \tag{5}$$

- The objective function is

$$s_n(Z_n; \theta) = m_n'(Z_n; \theta) W(\hat{\tau}_n) m_n'(Z_n; \theta) \tag{6}$$

The GMM estimator

- Error functions are of the form

$$\varepsilon(y_t, x_t; \theta) = y_t - \phi(x_t; \theta), \tag{3}$$

- Moment conditions are defined by interacting a vector of instrumental variables $z(x_t)$ with error functions:

$$m(y_t, x_t; \theta) = z(x_t) \otimes \varepsilon(y_t, x_t; \theta) \tag{4}$$

- Average moment conditions are

$$m_n(Z_n; \theta) = \frac{1}{n} \sum_{t=1}^{n} m(y_t, x_t; \theta) \tag{5}$$

- The objective function is

$$s_n(Z_n; \theta) = m_n'(Z_n; \theta) W(\hat{\tau}_n) m_n'(Z_n; \theta) \tag{6}$$

The GMM estimator

- Error functions are of the form

$$\varepsilon(y_t, x_t; \theta) = y_t - \phi(x_t; \theta), \tag{3}$$

- Moment conditions are defined by interacting a vector of instrumental variables $z(x_t)$ with error functions:

$$m(y_t, x_t; \theta) = z(x_t) \otimes \varepsilon(y_t, x_t; \theta) \tag{4}$$

- Average moment conditions are

$$m_n(Z_n; \theta) = \frac{1}{n} \sum_{t=1}^{n} m(y_t, x_t; \theta) \tag{5}$$

- The objective function is

$$s_n(Z_n; \theta) = m_n'(Z_n; \theta) W(\hat{\tau}_n) m_n'(Z_n; \theta) \tag{6}$$

The SNM estimator

- Consider a long simulation from the DLV model, $\widetilde{Z}_S$.
- Kernel regression may be used to fit $\phi(x_t;\theta)$, using this simulated data

$$\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) = \sum_{s=1}^{S} \widetilde{w}_s \widetilde{y}_s(\theta) \tag{7}$$

- the weight $\widetilde{w}_s$ is

$$\widetilde{w}_s = \frac{K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)}{\sum_{s=1}^{S} K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)} \tag{8}$$

- $\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) \xrightarrow{a.s.} \phi(x_t,\theta)$, for almost all x_t, as $S \to \infty$.
- S can be made as large as we like! (Jump back to last slide)

The SNM estimator

- Consider a long simulation from the DLV model, $\widetilde{Z}_S$.
- Kernel regression may be used to fit $\phi(x_t;\theta)$, using this simulated data

$$\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) = \sum_{s=1}^{S} \widetilde{w}_s \widetilde{y}_s(\theta) \tag{7}$$

- the weight $\widetilde{w}_s$ is

$$\widetilde{w}_s = \frac{K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)}{\sum_{s=1}^{S} K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)} \tag{8}$$

- $\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) \stackrel{a.s.}{\to} \phi(x_t,\theta)$, for almost all x_t, as $S \to \infty$.
- S can be made as large as we like! (Jump back to last slide)

The SNM estimator

- Consider a long simulation from the DLV model, $\widetilde{Z}_S$.
- Kernel regression may be used to fit $\phi(x_t;\theta)$, using this simulated data

$$\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) = \sum_{s=1}^{S} \widetilde{w}_s \widetilde{y}_s(\theta) \qquad (7)$$

- the weight $\widetilde{w}_s$ is

$$\widetilde{w}_s = \frac{K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)}{\sum_{s=1}^{S} K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)} \qquad (8)$$

- $\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) \overset{a.s.}{\to} \phi(x_t,\theta)$, for almost all x_t, as $S \to \infty$.
- S can be made as large as we like! (Jump back to last slide)

The SNM estimator

- Consider a long simulation from the DLV model, $\widetilde{Z}_S$.
- Kernel regression may be used to fit $\phi(x_t;\theta)$, using this simulated data

$$\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) = \sum_{s=1}^{S} \widetilde{w}_s \widetilde{y}_s(\theta) \tag{7}$$

- the weight $\widetilde{w}_s$ is

$$\widetilde{w}_s = \frac{K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)}{\sum_{s=1}^{S} K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)} \tag{8}$$

- $\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) \overset{a.s.}{\to} \phi(x_t,\theta)$, for almost all x_t, as $S \to \infty$.
- S can be made as large as we like! (Jump back to last slide)

The SNM estimator

- Consider a long simulation from the DLV model, $\widetilde{Z}_S$.
- Kernel regression may be used to fit $\phi(x_t;\theta)$, using this simulated data

$$\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) = \sum_{s=1}^{S} \widetilde{w}_s \widetilde{y}_s(\theta) \quad (7)$$

- the weight $\widetilde{w}_s$ is

$$\widetilde{w}_s = \frac{K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)}{\sum_{s=1}^{S} K\left(\frac{x_t - \widetilde{x}_s(\theta)}{h_S}\right)} \quad (8)$$

- $\widetilde{\phi}_S(x_t;\widetilde{Z}_S(\theta)) \stackrel{a.s.}{\to} \phi(x_t,\theta)$, for almost all x_t, as $S \to \infty$.
- S can be made as large as we like! (Jump back to last slide)

The SNM estimator *is* the GMM estimator if S is large enough

Classical linear model

$$\text{Linear Model:} \begin{cases} y & = \beta_1 + \beta_2 x + \varepsilon \\ x & \sim U(0,1) \\ \varepsilon & \sim N(0,1) \end{cases} \tag{9}$$

- SNM estimation with $S = 500000$, 1000 Monte Carlo reps

The SNM estimator *is* the GMM estimator if S is large enough

Classical linear model

$$\text{Linear Model: } \begin{cases} y & = \beta_1 + \beta_2 x + \varepsilon \\ x & \sim U(0,1) \\ \varepsilon & \sim N(0,1) \end{cases} \quad (9)$$

- SNM estimation with $S = 500000$, 1000 Monte Carlo reps

...continuation

- $$\underset{}{\widehat{\beta}_1(SNM)} = -\underset{(0.00023012)}{0.00106912} + \underset{(0.00030566)}{1.00292}\,\widehat{\beta}_1(GMM) - \underset{(0.00050332)}{0.00267236}\,\beta_1$$

  $$T = 1000 \quad \bar{R}^2 = 0.9999 \quad F(2,997) = 8.5632\text{e+6} \quad \hat{\sigma} = 0.0036169$$

  (standard errors in parentheses)

- $$\widehat{\beta}_2(SNM) = \underset{(0.00038392)}{2.50475\text{e-5}} + \underset{(0.00029626)}{1.00389}\,\widehat{\beta}_2(GMM) - \underset{(0.00073023)}{0.000178451}\,\beta_2$$

  $$T = 1000 \quad \bar{R}^2 = 0.9999 \quad F(2,997) = 6.9636\text{e+6} \quad \hat{\sigma} = 0.0061481$$

  (standard errors in parentheses)

...continuation

- $$\widehat{\beta}_1(SNM) = \underset{(0.00023012)}{-0.00106912} + \underset{(0.00030566)}{1.00292}\, \widehat{\beta}_1(GMM) - \underset{(0.00050332)}{0.00267236}\, \beta_1$$
  $$T = 1000 \quad \bar{R}^2 = 0.9999 \quad F(2,997) = 8.5632\text{e+}6 \quad \hat{\sigma} = 0.0036169$$
  (standard errors in parentheses)

- $$\widehat{\beta}_2(SNM) = \underset{(0.00038392)}{2.50475\text{e-}5} + \underset{(0.00029626)}{1.00389}\, \widehat{\beta}_2(GMM) - \underset{(0.00073023)}{0.000178451}\, \beta_2$$
  $$T = 1000 \quad \bar{R}^2 = 0.9999 \quad F(2,997) = 6.9636\text{e+}6 \quad \hat{\sigma} = 0.0061481$$
  (standard errors in parentheses)

Estimating the optimal weight matrix

- The asymptotic covariance matrix of the moment conditions is

$$\Omega = \lim_{n\to\infty} \mathscr{E}\left[n\widetilde{m}_n(\theta^0)\widetilde{m}_n(\theta^0)'\right] \tag{10}$$

- To do inference, or to do a second round of estimation using the efficient weight matrix, we need to estimate Ω.
- This is usually done using HAC covariance estimators along the lines of Newey and West (1987).
- Note that the moment conditions $\widetilde{m}_n()$ themselves can be simulated as many times as we like, given an initial estimate of θ
- The sample covariance of a large number of draws of $\sqrt{n}\widetilde{m}_n()$ is a simple estimator that seems to work very well in preliminary work. Has this been used previously?

Estimating the optimal weight matrix

- The asymptotic covariance matrix of the moment conditions is

$$\Omega = \lim_{n\to\infty} \mathscr{E}\left[n\widetilde{m}_n(\theta^0)\widetilde{m}_n(\theta^0)'\right] \tag{10}$$

- To do inference, or to do a second round of estimation using the efficient weight matrix, we need to estimate Ω.
- This is usually done using HAC covariance estimators along the lines of Newey and West (1987).
- Note that the moment conditions $\widetilde{m}_n()$ themselves can be simulated as many times as we like, given an initial estimate of θ
- The sample covariance of a large number of draws of $\sqrt{n}\widetilde{m}_n()$ is a simple estimator that seems to work very well in preliminary work. Has this been used previously?

Estimating the optimal weight matrix

- The asymptotic covariance matrix of the moment conditions is

$$\Omega = \lim_{n\to\infty} \mathscr{E}\left[n\widetilde{m}_n(\theta^0)\widetilde{m}_n(\theta^0)'\right] \tag{10}$$

- To do inference, or to do a second round of estimation using the efficient weight matrix, we need to estimate Ω.
- This is usually done using HAC covariance estimators along the lines of Newey and West (1987).
- Note that the moment conditions $\widetilde{m}_n()$ themselves can be simulated as many times as we like, given an initial estimate of θ
- The sample covariance of a large number of draws of $\sqrt{n}\widetilde{m}_n()$ is a simple estimator that seems to work very well in preliminary work. Has this been used previously?

Estimating the optimal weight matrix

- The asymptotic covariance matrix of the moment conditions is

$$\Omega = \lim_{n\to\infty} \mathscr{E}\left[n\widetilde{m}_n(\theta^0)\widetilde{m}_n(\theta^0)'\right] \quad (10)$$

- To do inference, or to do a second round of estimation using the efficient weight matrix, we need to estimate Ω.
- This is usually done using HAC covariance estimators along the lines of Newey and West (1987).
- Note that the moment conditions $\widetilde{m}_n()$ themselves can be simulated as many times as we like, given an initial estimate of θ
- The sample covariance of a large number of draws of $\sqrt{n}\widetilde{m}_n()$ is a simple estimator that seems to work very well in preliminary work. Has this been used previously?

Estimating the optimal weight matrix

- The asymptotic covariance matrix of the moment conditions is

$$\Omega = \lim_{n\to\infty} \mathscr{E}\left[n\widetilde{m}_n(\theta^0)\widetilde{m}_n(\theta^0)'\right] \tag{10}$$

- To do inference, or to do a second round of estimation using the efficient weight matrix, we need to estimate Ω.
- This is usually done using HAC covariance estimators along the lines of Newey and West (1987).
- Note that the moment conditions $\widetilde{m}_n()$ themselves can be simulated as many times as we like, given an initial estimate of θ
- The sample covariance of a large number of draws of $\sqrt{n}\widetilde{m}_n()$ is a simple estimator that seems to work very well in preliminary work. Has this been used previously?

Comparisons to other estimators

- Monte Carlo studies, taking advantage of published results for other estimators
- Short simulations ($S = 5000$) to limit computational burden
- Inefficient weight matrix: $W_n = I$, again to limit computational burden
- 1000 replications used in all cases

Autoregressive Tobit

Fermanian and Salanié 2004

$$\text{AR Tobit:}\begin{cases} y_t = & \max(0, y_t^*) \\ y_t^* = & \alpha + \beta y_{t-1}^* + \sigma \varepsilon_t \\ \varepsilon_t \sim & IIN(0,1) \end{cases} \quad (11)$$

SNM estimation of AR Tobit

- four endogenous variables used to define error functions:
  - y_t (to provide information on α),
  - $y_t y_{t-1}$ and $y_t y_{t-2}$ (to provide information on β)
  - $y_t^2 - (\overline{y})^2$ (to provide information on σ).
- conditioning variables:
  - y_{t-1}
  - y_{t-2}.
- instruments: the conditioning variables, plus a vector of ones
- so a total of 12 moments are used to estimate 3 parameters
- simulation length $S = 5000$ (this is short to speed up Monte Carlo)

AR Tobit results

Table: AR Tobit Results, SNM

	Mean	Bias	St. Dev.	RMSE	RMSE NPSML
α	0.022	0.022	0.082	0.085	0.215
β	0.599	-0.099	0.111	0.148	0.151
σ	1.119	-0.119	0.166	0.204	0.264

Stochastic volatility (I)

Andersen, Chung and Sorensen (1999)

$$\text{SV1:} \begin{cases} y_t & = \exp\left(y_t^*/2\right)\varepsilon_{t,1} \\ y_t^* & = \alpha + \beta y_{t-1}^* + \sigma\varepsilon_{t,2} \end{cases} \tag{12}$$

Stochastic volatility (I)

- endogenous variables:
  - $100y_t^2$ (information on α and σ)
  - $100y_t^2 y_{t-1}^2$ (information on β)
- conditioning variables
  - y_{t-1}
  - y_{t-1}^2.
- instruments are conditioning variables and vector of ones
- so a total of 6 moments are used to estimate 3 parameters
- simulation length $S = 5000$ (this is short to speed up Monte Carlo)

SV1 results

Table: Stochastic Volatility Results, SV1, n=2000. RMSE for GMM/EMM is the lowest value reported by Andersen et al. (1999) in their Table 5, which reports RMSEs for a number of GMM and EMM models.

	Statistics for SNM			RMSE	
	Mean	Bias	St. Dev.	SNM	GMM/EMM
α	-0.750	-0.014	0.244	0.244	0.224
β	0.899	0.001	0.028	0.028	0.030
σ	0.407	-0.044	0.129	0.136	0.049

Stochastic volatility (II)

Fermanian and Salanié (2004), Altissimo and Mele (2007)

$$\text{SV2:} \begin{cases} y_t & = \sigma_b \exp\left(y_t^*/2\right)\varepsilon_{1t} \\ y_t^* & = \phi y_{t-1}^* + \sigma_\varepsilon \varepsilon_{2t} \end{cases} \quad (13)$$

SV2 results

Table: Stochastic Volatility Results, SV2. RMSE for NPSML calculated using information in Table 4 of Fermanian and Salanié (2004). RMSEs for CD-SNE and J-SNE taken from Altissimo and Mele (2007), Table 1.

	Statistics for SNM			RMSE			
	Mean	Bias	St. Dev.	SNM	NPSML	CD-SNE	J-SNE
ϕ	0.943	-0.007	0.017	0.018	0.107	0.110	0.095
σ_b	0.022	-0.003	0.002	0.004	0.004	0.003	0.005
σ_ε	0.206	-0.054	0.057	0.079	0.180	0.134	0.149

Factor ARCH

Billio and Monfort (2003)

$$\text{FA:} \begin{cases} y_t & = \beta y_t^* + \varepsilon_t \\ y_t^* & = \sqrt{h_t}\varepsilon_t^* \\ h_t & = \alpha_1 + \alpha_2 \left(y_{t-1}^*\right)^2 \end{cases} \quad (14)$$

In general, y_t is a vector, and y_t^* is a lower dimensional vector of latent factors. In this case, y_t has two elements and y_t^* is a scalar.

FA results

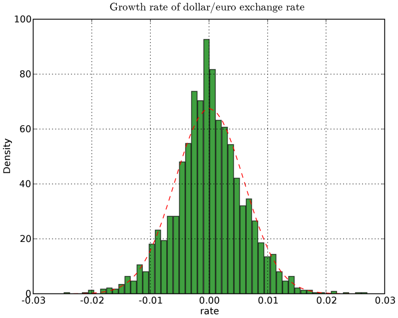
Table: Factor ARCH Results. RMSEs for best alternative estimators taken from Billio and Monfort (2003) Table 5.

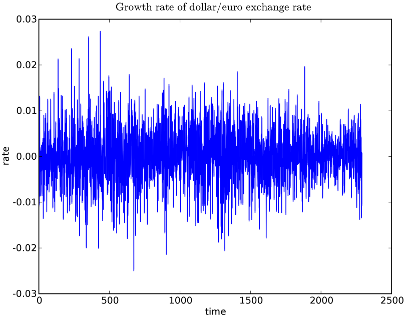
	Statistics for SNM			RMSE	
	Mean	Bias	St. Dev.	SNM	Best alternative
α_1	0.204	0.004	0.099	0.099	0.132
α_2	0.551	-0.149	0.168	0.224	0.309
σ	0.498	-0.002	0.064	0.064	0.141
β_2	-0.426	0.074	0.291	0.300	0.269

Application: Factor GARCH model for exchange rate data

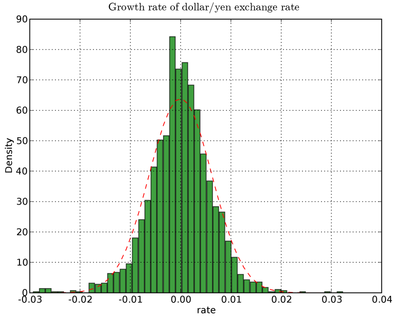
- The data are the growth rates (100$\times$log difference) of the daily spot \$/euro and \$/yen exchange rates at New York, noon.
- sample January 04, 1999 to February 12, 2008. There are 2291 observations.
- source `http://www.federalreserve.gov/releases/h10/Hist/`
- calendar effects (weekends and non-trading days) are **ignored**

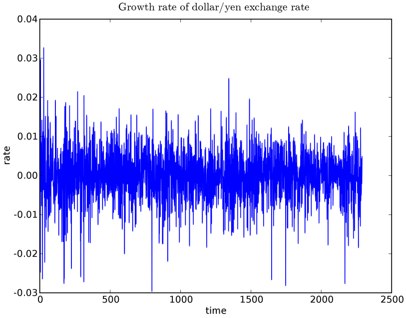
Dollar-Euro


Growth rate of dollar/euro exchange rate
Density
rate
100
80
60
40
20
0
-0.03
-0.02
-0.01
0.00
0.01
0.02
0.03


Growth rate of dollar/euro exchange rate
rate
time
0.03
0.02
0.01
0.00
-0.01
-0.02
-0.03
0
500
1000
1500
2000
2500

Dollar-Yen


Growth rate of dollar/yen exchange rate
Density
rate
90
80
70
60
50
40
30
20
10
0
-0.03
-0.02
-0.01
0.00
0.01
0.02
0.03
0.04


Growth rate of dollar/yen exchange rate
rate
time
0.04
0.03
0.02
0.01
0.00
-0.01
-0.02
-0.03
0
500
1000
1500
2000
2500

Factor GARCH

Billio and Monfort (2003)

$$\text{FG:}\begin{cases} y_t & = \beta y_t^* + \varepsilon_t \\ y_t^* & = \sqrt{h_t}\varepsilon_t^* \\ h_t & = \alpha_1 + \alpha_2\left(y_{t-1}^*\right)^2 + \delta h_{t-1} \end{cases}$$

- simple GARCH analogue of the factor ARCH model above.
- In general, y_t is a vector, and y_t^* is a lower dimensional vector of latent factors.
- In this case, y_t has two elements and y_t^* is a scalar.

SNM estimation

- endogenous variables:
  - $10y_t^2$ (information on α and σ)
  - $y_t^2 y_{t-1}^2$ (information on β)
  - $y_{t1}y_{t2}$
- conditioning variables
  - y_{t-1}
  - y_{t-1}^2.
- instruments are conditioning variables and vector of ones
- so a total of 25 moments are used to estimate 5 parameters
- simulation length $S = 100000$

SNM Results for FG model of exchange rates

```
*******************************************************
GMM estimation Factor GARCH

GMM Estimation Results
BFGS convergence: Normal convergence

Objective function value: 0.033428
Observations: 2289

               Value         df     p-value
X^2 test      76.517     20.000       0.000

      estimate     st. err      t-stat     p-value
a1       0.015       0.004       4.320       0.000
a2       0.146       0.023       6.239       0.000
sig      0.222       0.008      26.655       0.000
b2      -0.985       0.034     -29.351       0.000
delta    0.737       0.052      14.083       0.000
*******************************************************
```

Summary and comments

- SNM is about as efficient as EMM, which is "state of the art", at least for general DLV models
- SNM is more efficient than methods other than EMM
- SNM does not require estimating an auxiliary model, as is needed to use EMM. Doing this successfully may be difficult.
- New method for estimating efficient weight matrix - seems to work very well (preliminary)
- SNM *is* computationally demanding.
- As with ordinary GMM estimators, SNM objective function is not globally convex

Future work

- dimension reduction techniques, both for endogenous variables and conditioning variables
- application to agent-based models
- data-driven methods for choosing window width
- alternative nonparametric fitting methods - support vector machines (SVMs)? Approximate nearest neighbors (ANN)?
- how well does the method of estimating the covariance of the moments work?