VARs for Big Data

Big data

- Big Data is becoming ever more common in macroeconomics
- VAR (or factor) modeler should take this into account
- E.g. St Louis FRED database had 5,000 series in 2010, 20,000 in 2011, and currently 264,000 time series
- With interlinkages between countries, multi-country/panel VARs getting huge
- Mixed frequency (e.g. combining monthly and quarterly variables) leads to large VARs
- etc.

This Lecture:

- We are going to examine how Bayesian methods can be used to exploit information when modelling jointly big data (as a VAR)
- Two main directions:
    1. Methods that compress the data
    2. Methods that shrink coefficients
- First approach takes a large matrix X and projects it to a lower dimensional matrix F (factor and random compression methods)
- Second approach takes the parameters of a large model and shrinks irrelevant coefficients to zero (prior shrinkage, model selection, model averaging methods)
- I will also discuss how to combine big data with state space methods in order to add to our large econometric model features of interest (e.g. time-varying parameters)

Constant Parameter Models for Big Data

VAR with Bayesian shrinkage

- If our large matrix of “big data” follows a VAR, then things can get really messy
- E.g. a VAR(12) for 4 variables has 192 AR coefficients, a VAR(12) for 20 variables has 4800 coefficients
- One solution is to use hierarchical prior (e.g. SSVS or Dirichlet-Laplace)
- But when X_t is really large MCMC methods are too computationally demanding
- Minnesota (or natural conjugate) priors do not require use of MCMC methods and are feasible
- Banbura, Giannone, Reichlin (2010, JAE) showed that one can estimate 132-variable VAR(13) with Minnesota prior
- But such priors are subjective and extension to TVP or SV not computationally feasible (MCMC is required)

VAR with Bayesian shrinkage 2

- BGR (2010) have to shrink more than 200,000 coefficients!
- It is hard to believe that the likelihood has so much information in order to shrink to the correct direction so many coefficients
- Remember that when using big data and have many coefficients, estimation variance (uncertainty) is huge
- So in huge systems, shrinking ANY coefficient (even relevant ones) will always reduce the variance a lot
- Despite some increase in bias, the total error of fit (mse = bias + variance) will be lower
- Additionally, even the analytical posterior results on a 132 variable VAR can result in numerical instabilities

VAR with Bayesian shrinkage 3

- All the hierarchical shrinkage priors of Topic 2 (SSVS, LASSO, Dirichlet-Laplace etc) can be used in large systems
- ... at least this holds in theory, because in practice they rely on computationally intensive MCMC methods
- This is what makes the Minnesota-type prior based on the natural conjugate prior so attractive
- Shrinkage priors that rely on MCMC can usually allow estimation of VAR systems with up to (approximately) 50 variables
- Many authors focus on transforming the VAR in various ways in order to be able to estimate it in a computationally efficient way
- Names doing interested Bayesian research in this field include Andrea Carriero, Massimiliano Marcelino, Todd Clark, Joshua Chan

Bayesian Compressed VAR

- Recently, Koop, Korobilis and Pettnuzzo (2016, JOE) have proposed to use “compressive sensing” in the VAR
- Main idea:
  - Compress the VAR regressors through random projection
  - Use BMA to average across different random projections

- This is a machine learning method
- They apply Bayesian compressed VARs to forecast a 130-variable VARs with 13 lags (similar to Banbura et al (2010)), with more than $200,000$ parameters to estimate
  - Find good forecasting performance, relative to a host of alternative methods including DFM, FAVAR, and BVAR with Minnesota priors

Random Projection vs. Principal Component Analysis

- Random Projection (RP) is a projection method similar to Principal Component Analysis (PCA)
  - High-dimensional data is projected onto a low-dimensional subspace using a random matrix, whose columns have unit length
  - Unlike PCA, “loadings” are not estimated from data, rather generated randomly (“Data Oblivious” method)
- Inexpensive in terms of time/space. Random projection can be generated without even seeing the data
- Theoretical results show that RP preserves volumes and affine distances, or the structure of data (e.g., clustering)
  - Johnson-Lindenstrauss (1984) lemma: Any n point subset of Euclidean space can be embedded in $k = O\left(\log n/\epsilon^2\right)$ dimensions without distorting the distances between any pair of points by more than a factor of $1 \pm \epsilon$, for any $0 < \epsilon < 1$

Bayesian Compressed Regression (BCR)

- Start with the case of a scalar dependent variable y_t, $t = 1, ..., T$, predictor matrix $x_t = (x_{t,1}, ..., x_{t,k})'$, and linear regression model

$$y_t = x_t'\beta + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$$

  When $k \gg T$, estimation is either impossible (e.g. MLE), or computationally very hard (e.g. Bayesian regression with SSVS prior)
- Guhaniyogi and Dunson (2015, JASA) consider a compressed regression specification

$$y_t = (\Phi x_t)' \beta^c + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$$

  where Φ is an $(m \times k)$ compression matrix with $m \ll k$
- Conditional on Φ, estimating β^c and forecasting y_{t+1} is now very straightforward, and can be carried out using standard (Bayesian) regression methods

Projection matrix

- The elements $\{\Phi_{ij}\}$ can be generated quickly, e.g.

$$\Phi_{ij} \sim \mathcal{N}(0, 1)$$

Alternatively, Achlioptas (2003) use a sparse random projection

$$\Phi_{ij} = \begin{cases} -\sqrt{\varphi} & \text{with probability} & 1/2\varphi \\ 0 & \text{with probability} & 1 - 1/\varphi \\ \sqrt{\varphi} & \text{with probability} & 1/2\varphi \end{cases}$$

where $\varphi = 1$ or 3.

- We follow the scheme of Guhaniyogi and Dunson (2015)

$$\Phi_{ij} = \begin{cases} -\frac{1}{\sqrt{\varphi}} & \text{with probability} & \varphi^2 \\ 0 & \text{with probability} & 2(1-\varphi)\varphi \\ \frac{1}{\sqrt{\varphi}} & \text{with probability} & (1-\varphi)^2 \end{cases}$$

where $\varphi \in (0.1, 0.9)$ and is estimated from the data

Model Averaging

- Guhaniyogi and Dunson (2015) show that BCR produces a predictive density for y_{t+1} that (under mild conditions) converges to its true predictive density (large k, small T asymptotics)
- To limit sensitivity of results to choice of m and φ, generate R random compressions based on different (m, φ) pairs.
- Use BMA to integrate out (m, φ) from predictive density of y_{t+1}:

$$p\left(y_{t+1}|\mathcal{Y}^t\right) = \sum_{r=1}^{R} p\left(y_{t+1}|M_r, \mathcal{Y}^t\right) p\left(M_r|\mathcal{Y}^t\right)$$

  where $p\left(M_r|\mathcal{Y}^t\right)$ denotes model M_r posterior probability and M_r denotes the r-th pair of (m, φ) values, where:

  - $\varphi \sim \mathcal{U}\left(0.1, 0.9\right)$
  - $m \sim \mathcal{U}\left(2\ln\left(k\right), \min\left(T, k\right)\right)$

Large VAR setup

- VAR(p) for $n \times 1$ vector of dependent variables is :

$$Y_t = a_0 + \sum_{j=1}^{p} A_j Y_{t-j} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \Omega)$$

  Rewrite this compactly as

$$Y_t = BX_t + \varepsilon_t$$

  where B is an $n \times k$ matrix of coefficients, X_t is $k \times 1$, and $k = np + 1$. Also, note that Ω has $n(n+1)/2$ free parameters

- Potentially, many parameters to estimate. E.g., when $n = 130$ and $p = 13$, B has $220,000+$ parameters to estimate, while Ω has $8,500+$ unconstrained elements

Bayesian Compressed VAR (BCVAR)

- Define the Compressed VAR as

$$Y_t = B^c \left(\Phi X_t \right) + \varepsilon_t$$

  where the projection matrix Φ is $m \times k$, $m \ll k$
- Conditional on a given Φ (its elements randomly drawn as before), estimation and forecasts for the compressed VAR above are trivial and very fast to compute
- Note:
  - h-step ahead forecasts (for $h > 1$) not available analytically. For those, rewrite compressed VAR as

$$Y_t = \left(B^c \Phi \right) X_t + \varepsilon_t$$

    and iterate forward in the usual way
  - The compressed VAR above imposes the same compression (ΦX_t) in all equations; may be too restrictive
  - So far, no compression is applied to the elements of Ω

Compressing the VAR covariance matrix

- Ω has $n\,(n+1)\,/2$ unconstrained elements, so we modify the BCVAR to allow also for their compression
- Use a triangular decomposition of Ω

$$A\Omega A' = \Sigma\Sigma,$$

Σ is a diagonal matrix with diagonal elements σ_i
A is a lower triangular matrix with ones on the diagonal

- Define $A = I + \widetilde{A}$, where $\widetilde{A}$ is lower triangular but with zeros on the diagonal, and rewrite uncompressed VAR as

$$\begin{aligned} Y_t &= \Gamma X_t + \widetilde{A}\,(-Y_t) + \Sigma E_t \\ &= \Theta Z_t + \Sigma E_t \end{aligned}$$

where $E_t \sim \mathcal{N}\,(0, I_n)$, $Z_t = [X_t, -Y_t]$ and $\Theta = \left[\Gamma, \widetilde{A}\right]'$

Compressing the VAR covariance matrix

- Compression can be accomplished as follows:

$$Y_t = \Theta^c \left(\Phi Z_t \right) + \Sigma E_t$$

  where Φ is now an $m \times (k + n)$ random compression matrix
  Note that we would still be relying on the same compression matrix (Φ) for all equations

- Alternatively, we can allow each equation to have its own random compression matrix (of size $m_i \times (k + i - 1)$):

$$Y_{i,t} = \Theta_i^c \left(\Phi_i Z_{i,t} \right) + \sigma_i E_{i,t}$$

  Having n compression matrices (each of different dimension and with different randomly drawn elements) allows for the explanatory variables of different equations to be compressed in potentially different ways

Estimation and Prediction

- Estimation is performed equation-by-equation, conditional on a known (generated) Φ_i
- We choose a standard natural conjugate prior:

$$
\begin{aligned}
\Theta_i^c &\sim \mathcal{N}\left(\underline{\Theta}_i^c, \sigma_i^2 \underline{V}_i\right) \\
\sigma_i^{-2} &\sim \mathcal{G}\left(\underline{s}^{-2}, \underline{\nu}\right)
\end{aligned}
$$

  where $i = 1, ..., n$.
  Posterior location and scale parameters for $\Theta_i^c, \sigma_i^{-2}$ are available analytically
- 1-step ahead forecasts are also available analytically
- h-step ahead forecasts (for $h > 1$) require some extra work
  Rewrite compressed VAR as

$$
Y_{i,t} = \left(\Theta_i^c \Phi_i\right) Z_{i,t} + \sigma_i E_{i,t}
$$

  and iterate forward in the usual way, one equation at a time

Model averaging

- We generate many random $\Phi^{(r)}$ (or $\Phi_i^{(r)}$), $r = 1, ..., R$ based on different (m, φ) pairs, then implement BMA as follows
- First, we rely on BIC instead of the marginal likelihood. We compute model M_r BIC as

$$BIC_r = \ln\left(\left|\overline{\Sigma}_r\right|\right) + \frac{\ln(t)}{t}\left(n \times \sum_{i=1}^{n} m_i\right)$$

Posterior model probability is approximated by

$$\Pr\left(M_r|\mathcal{Y}^t\right) \approx \frac{\exp\left(-\frac{1}{2}BIC_r\right)}{\sum_{\varsigma=1}^{R}\exp\left(-\frac{1}{2}BIC_\varsigma\right)}$$

- Next,

$$p\left(Y_{t+h}|\mathcal{Y}^t\right) = \sum_{r=1}^{R} p\left(Y_{t+h}|M_r, \mathcal{Y}^t\right) p\left(M_r|\mathcal{Y}^t\right)$$

where $h = 1, ..., H$

Empirical Application

- We use the “FRED-MD” monthly macro data (McCracken and Ng, 2015), 2015-05 vintage
  - 134 series covering: (1) the real economy (output, labor, consumption, orders and inventories), (2) money and prices, (3) financial markets (interest rates, exchange rates, stock market indexes).
- Series are transformed as in Banbura et al (2010) by applying logarithms, excepts when series are already expressed in rates
- Final sample is 1960M3 - 2014M12 (658 obs.)
- We focus on forecasting: Employment (PAYEMS), Inflation (CPIAUCSL), Federal fund rate (FEDFUNDS), Industrial production (INDPRO), Unemployment rate (UNRATE), Producer Price Index (PPIFGS), and 10 year US Treasury Bond yield (GS10).

VAR specifications

- We have three sets of VARs: **Medium**, **Large**, and **Huge**
- All VARs include seven key variables of interest: Employment, Inflation, Fed Fund rate, IPI, Unemployment, PPI, and 10 yr bond yield
- **Medium** VAR has 19 variables - similar to Banbura et al (2010)
- **Large** VAR has 46 variables - similar to Carriero et al (2011)
- **Huge** VAR has 129 variables
- Note: All four VARs produce forecasts for the variables of interest, but imply different information sets

Forecast evaluation

- We forecast $h = 1$ to 12 months ahead
- Initial estimation based on first half of the sample, $t = 1, ..., T_0$; forecast evaluation over the remaining half, $t = T_0 + 1, ..., T - h$ ($T_0 = 1987M7$, $T = 2014M12$)
- Forecasts are computed recursively, using an expanding estimation window.
- We evaluate forecasts relative to an AR(1) benchmark and focus on
  - Mean squared forecast error (MSFE)
  - Cumulative sum of squared forecast errors (Cum SSE)
  - Average (log) predictive likelihoods (ALPLs)
- Competing methods are **DFM** using PCA as in Stock and Watson (2002), **FAVAR** using PCA as in Bernanke et al (2005) with selection of lags and factors using BIC, and **BVAR** with Minnesota prior as in Banbura et al (2010)

Relative MSFE ratios, Large VAR

Variable	*DFM*	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*	*DFM*	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*
			$h = 1$					$h = 2$		
PAYEMS	1.273	0.906	**0.788****	0.888***	0.905**	0.897	0.698***	**0.521*****	0.805***	0.837***
CPIAUCSL	1.129	1.100	1.009	0.998	**0.953**	1.160	1.118	1.110	0.943	**0.909****
FEDFUNDS	2.387	1.671	2.461	**1.093**	1.103	2.133	1.415	2.594	**0.991**	1.150
INDPRO	0.870*	0.890**	**0.783*****	0.838***	0.914***	0.861	0.966	**0.772****	0.934*	0.929*
UNRATE	30.779	**0.786*****	0.824*	0.798***	0.855**	14.267	**0.661*****	0.666*	0.722***	0.758**
PPIFGS	1.042	1.013	1.045	0.987	**0.986**	1.151	1.056	1.162	1.012	**1.004**
GS10	1.001	**0.983**	1.106	1.006	0.992	1.029	**0.987**	1.149	1.052	1.044
			$h = 3$					$h = 6$		
PAYEMS	0.814	0.710***	**0.489*****	0.757***	0.748***	0.823	0.832**	**0.647***	0.773**	0.762***
CPIAUCSL	1.134	1.067	1.154	0.948	**0.913****	1.055	0.970	0.999	0.910**	**0.902****
FEDFUNDS	1.878	**1.026**	2.241	1.034	1.108	1.433	**0.944**	1.224	1.098	1.088
INDPRO	0.925	0.943	**0.862**	0.957*	0.952	**0.942**	0.962	0.980	0.959	0.984
UNRATE	8.547	0.631***	**0.615***	0.677***	0.731**	3.442	0.663***	**0.617**	0.671***	0.712**
PPIFGS	1.163	1.018	1.177	1.021	**1.013**	1.129	1.014	1.095	1.012	**0.998**
GS10	1.048	**1.030**	1.222	1.057	1.059	1.040	**1.018**	1.115	1.043	1.029
			$h = 9$					$h = 12$		
PAYEMS	0.895	0.930	**0.840**	0.865	0.844**	**0.919**	0.966	0.999	0.972	0.940
CPIAUCSL	1.055	0.975	0.932	0.887**	**0.867*****	1.065	0.984	0.904	0.902**	**0.879*****
FEDFUNDS	1.250	**0.999**	1.139	1.060	1.028	1.131	**0.994**	1.259	1.092	1.041
INDPRO	0.983	**0.972**	1.018	1.004	0.999	**0.954**	0.979	1.056	1.002	1.020
UNRATE	2.129	**0.684*****	0.715	0.698**	0.739**	1.595	**0.710*****	0.831	0.728**	0.756**
PPIFGS	1.068	1.000	1.051	**0.985**	0.992	1.101	1.002	1.039	1.004	**0.972**
GS10	1.008	**1.001**	1.050	1.009	1.019	1.015	**1.001**	1.054	1.023	1.014

Average (log) predictive likelihoods, Large VAR

Variable	*DFM*	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*	*DFM*	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*
			$h=1$					$h=2$		
PAYEMS	0.038	0.107***	**0.259*** **	0.065***	0.063***	0.130***	0.168***	**0.399*** **	0.120***	0.113***
CPIAUCSL	-0.401	-0.239	-0.775	-0.078	**0.104**	-0.916	-0.581	-2.312	-0.222	**-0.220**
FEDFUNDS	0.017	0.060***	**0.149*** **	-0.017	-0.018	-0.016	**0.021** **	-0.023	-0.004	-0.012
INDPRO	-0.050	-0.034	-0.059	-0.011	**0.045*** **	0.086	0.105	**0.240** **	0.102*	0.146
UNRATE	-1.902	**0.138*** **	0.122**	0.096***	0.061***	-1.313	0.144**	**0.235** **	0.169***	0.144**
PPIFGS	-0.025	-0.030	-0.711	-0.023	**0.028**	-0.567	-0.213	-1.207	**0.023**	-0.144
GS10	**0.042***	0.036	-0.012	0.001	0.011	-0.011	**0.003**	-0.010	-0.021	-0.027
			$h=3$					$h=6$		
PAYEMS	0.141***	0.141***	**0.407*** **	0.148***	0.149***	0.116**	0.068***	**0.282*** **	0.114***	0.157***
CPIAUCSL	-0.588	**-0.232**	-1.949	-0.446	-0.269	0.010	**0.048**	-0.889	-0.189	-0.002
FEDFUNDS	-0.042	0.027***	**0.032**	0.001	-0.001	-0.019	0.007*	**0.158*** **	-0.013	-0.014
INDPRO	0.073*	0.199	0.044	**0.223**	0.074*	**0.060***	-0.053	-0.294	-0.045	-0.081
UNRATE	-1.001	0.116	0.356	**0.357***	0.340*	-0.078	0.495	0.502	**0.958**	0.869
PPIFGS	-0.431	**0.004**	-1.109	-0.020	-0.009	-0.151	-0.108	-0.857	-0.116	**-0.060**
GS10	0.001	**0.002**	-0.025	-0.002	-0.029	-0.006	**0.002**	-0.009	-0.021	-0.024
			$h=9$					$h=12$		
PAYEMS	0.067*	0.027*	**0.105***	0.065***	0.099***	**0.062***	0.039	0.016	0.041**	0.024
CPIAUCSL	-0.249	**0.027**	-0.943	-0.081	-0.228	**-0.027**	-0.127	-0.784	-0.106	-0.053
FEDFUNDS	-0.011	-0.006	**0.148*** **	-0.024	-0.022	-0.003	-0.005	**0.136*** **	-0.028	-0.016
INDPRO	0.105	**0.122**	-0.168	0.092	-0.002	**0.160**	-0.057	-0.231	-0.109	0.058
UNRATE	0.892	**1.495**	0.180	1.326	1.136	1.499	**1.878**	-0.016	1.367	1.040
PPIFGS	-0.165	0.014	-0.629	0.029	**0.061**	**-0.038**	-0.185	-0.711	-0.150	-0.145
GS10	-0.012	-0.011	**0.024**	-0.022	-0.024	-0.034	-0.008	**0.010**	-0.018	-0.040

Forecast evaluation

- Multivariate measure of forecast performance is

$$we_{i,\tau+h} = \left(e'_{i,\tau+h} \times W \times e_{i,\tau+h}\right)$$

$e_{i,\tau+h} = Y_{\tau+h} - \widehat{Y}_{i,\tau+h}$ is the $(N \times 1)$ vector of forecast errors, and W is an $(N \times N)$ matrix of weights

- We set the matrix W to be a diagonal matrix featuring on the diagonal the inverse of the variances of the series to be forecast
- Next, define

$$WMSFE_{ih} = \frac{\sum_{\tau=\underline{t}}^{\bar{t}-h} we_{i,\tau+h}}{\sum_{\tau=\underline{t}}^{\bar{t}-h} we_{bcmk,\tau+h}}$$

where $\underline{t}$ and $\bar{t}$ denote the start and end of the out-of-sample period

Forecast evaluation

- Finally, we consider the multivariate average log predictive likelihood differentials between model i and the benchmark AR(1),

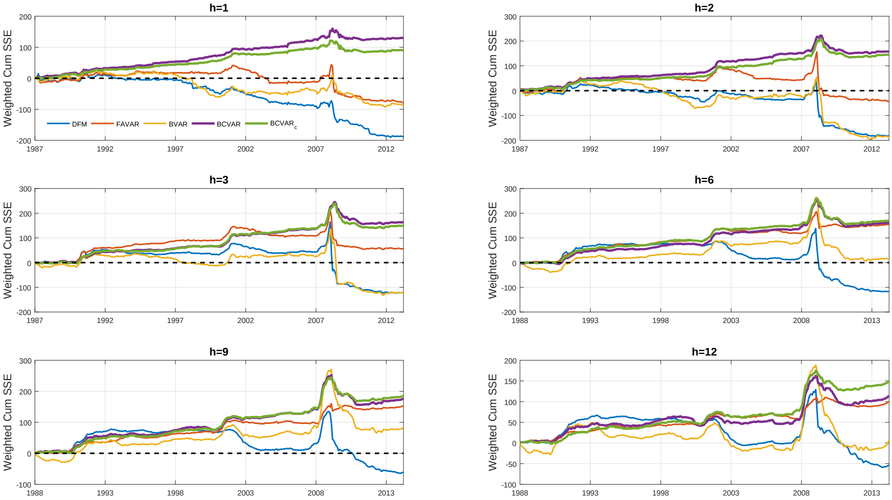
$$MVALPL_{ih} = \frac{1}{\bar{t} - \underline{t} - h + 1} \sum_{\tau=\underline{t}}^{\bar{t}-h} \left(MVLPL_{i,\tau+h} - MVLPL_{bcmk,\tau+h} \right),$$

  where:
  - $MVLPL_{i,\tau+h}$ denote the multivariate log predictive likelihoods of model i at time $\tau + h$
  - and $MVLPL_{bcmk,\tau+h}$ denote the multivariate log predictive likelihoods of the benchmark model at time $\tau + h$

Multivariate forecast comparisons

Fcst h.	*Medium VAR*									
	WTMSFE					*MVALPL*				
	DFM	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*	*DFM*	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*
h= 1	1.232	1.143	1.194	**0.936*** **	0.938***	-1.188	0.788***	**1.005*** **	0.919***	0.318***
h= 2	1.092	1.125	1.154	0.937*	**0.936** **	-0.590	0.912***	**1.222*** **	1.120***	0.514***
h= 3	1.066	1.051	1.082	0.949	**0.939* **	-0.277	1.053***	**1.362*** **	1.222***	0.575***
h= 6	1.035	0.961	1.005	**0.936**	0.938	0.255	1.216***	**1.472*** **	1.383***	0.690***
h= 9	1.019	0.934	0.973	**0.926* **	0.930*	0.638	1.336***	**1.502*** **	1.471***	0.703***
h=12	1.017	**0.941**	1.002	0.954	0.960	0.911	1.434**	1.425***	**1.465*** **	0.699***
	Large VAR									
	DFM	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*	*DFM*	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*
h= 1	1.288	1.080	1.172	**0.968**	0.975	-1.292	0.830***	**0.913*** **	0.827***	0.257***
h= 2	1.255	1.051	1.224	**0.961**	0.979	-0.685	0.970***	0.937***	**1.071*** **	0.323***
h= 3	1.211	0.969	1.192	**0.962**	0.964	-0.407	1.089***	1.024***	**1.159*** **	0.424***
h= 6	1.110	**0.949** **	0.994	0.954	0.950*	0.210	1.170***	**1.347*** **	1.280***	0.551***
h= 9	1.080	0.967*	0.988	0.953	**0.944* **	0.623	1.311***	1.337***	**1.381*** **	0.574***
h=12	1.061	0.971	1.033	0.980	**0.960**	0.906	**1.390*** **	1.134***	1.362***	0.476***
	Huge VAR									
	DFM	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*	*DFM*	*FAVAR*	*BVAR*	*BCVAR*	*BCVAR$_c$*
h= 1	1.117	1.050	1.055	**0.920*** **	0.944***	-0.342	**0.931*** **	0.760***	0.921***	0.272***
h= 2	1.097	1.023	1.098	**0.916** **	0.923**	0.123	1.148***	0.875***	**1.203*** **	0.468***
h= 3	1.061	0.971	1.061	**0.917* **	0.924*	0.375**	**1.282*** **	1.012***	1.276***	0.510***
h= 6	1.055	0.927*	0.993	0.924	**0.920**	0.813**	1.426***	1.018***	**1.483*** **	0.675***
h= 9	1.028	0.930**	0.962	0.919	**0.915* **	1.108**	1.542***	1.027***	**1.555*** **	0.737***
h=12	1.024	0.955*	0.997	0.949	**0.933**	1.301**	**1.631*** **	0.712	1.593***	0.645***

Weighted Cum. Sum SSE diffs (Huge VAR)



Time-varying Parameter Models for Big Data

TVP-VARs for Big Data

- Large systems might be more helpful during crises (when linkages in the economy become more complex)
- However, information in variables alone might not be sufficient to capture structural changes
- E.g. mid-80s “Great Moderation” occurred (pre-1984 variance of GDP was twice as high as post-1984 to 2007-8)
- Best way to model events such as the Great Moderation is to allow for volatility and intercepts to change
- Even in large VARs with lots of information (variables), we might still need to allow for structural instabilities

Large TVP-VAR

- Koop and Korobilis (2013) provide a solution to the issue of estimating large VARs with time-varying coefficients and stochastic volatility
- The problem is computational: Running MCMC and Kalman filter for 100,000+ coefficients (as arises in large VARs) is computationally impossible
- Remember TVP-VAR is state space model

$$\begin{aligned} y_t &= z_t\beta_t + \varepsilon_t \text{ , } \varepsilon_t \sim N(0, \Sigma_t) &(1)\\ \beta_t &= \beta_{t-1} + \eta_t \text{ , } \eta_t \sim N(0, Q) &(2)\end{aligned}$$

- If Q and Σ_t were known, MCMC not required
- State space methods (e.g. Kalman filter) all that is required for estimation

Large TVP-VAR

$$y_t = z_t\beta_t + \varepsilon_t \ , \varepsilon_t \sim N(0, \Sigma_t) \quad (3)$$

$$\beta_t = \beta_{t-1} + \eta_t \ , \eta_t \sim N(0, Q) \quad (4)$$

- Our solution is to estimate Q and Σ_t from past data in simple ways
- For the case of Σ_t we use a scheme popular in finance: Exponentially Weighted Moving Average (EWMA) filter
- This is the popular Riskmetrics model of the 1990s
- For Q we use a variance discounting scheme called forgetting
- In the next I will explain the details, and the simple updating scheme for the large TVP-VAR

$$
\begin{aligned}
y_t &= z_t\beta_t + \varepsilon_t \text{ , } \varepsilon_t \sim N(0, \Sigma_t) && (5)\\
\beta_t &= \beta_{t-1} + \eta_t \text{ , } \eta_t \sim N(0, Q) && (6)
\end{aligned}
$$

Kalman filter steps for state space model are

Predict step:

$$
\begin{aligned}
\beta_{t|t-1} &= \beta_{|t-1t-1} && (7)\\
P_{t|t-1} &= P_{t-1|t-1} + Q && (8)
\end{aligned}
$$

Update step:

$$
\begin{aligned}
v_{t|t-1} &= y_t - z_t\beta_{t|t-1} && (9)\\
S_t &= (\Sigma_t + z_t P_{t|t-1} z_t')^{-1} && (10)\\
K_t &= P_{t|t-1} z_t' S_t && (11)\\
\beta_{t|t} &= \beta_{t|t-1} + K_t v_{t|t-1} && (12)\\
P_{t|t} &= P_{t|t-1} + K_t z_t P_{t|t-1}' && (13)
\end{aligned}
$$

- So the covariances show up in two places:
  1. $P_{t|t-1} = P_{t-1|t-1} + Q$
  2. $S_t = (\Sigma_t + z_t P_{t|t-1} z_t')^{-1}$
- In the first equation we need data to estimate Q
- However, likelihood-based estimators (e.g. ML or Bayesian) will be based on the following Residual Sum of Squares

$$\sum_{t=1}^{T} (\beta_t - \beta_{t-1})'(\beta_t - \beta_{t-1}) \tag{14}$$

- This RSS is based on the latent quantity β_t and not on data matrices y_t and x_t (at least in an explicit way, implicitly it does)
- This is the reason that we go Bayesian in the TVP-VAR: we can use priors to "control" the behaviour of Q even when information in the data is weak
- This problem with Q was recognized early in the engineering literature

- In the case of the large VAR Q is massive. So we are going to replace it with "past data" (variance discounting) for two reasons
- One is estimation accuracy and numerical stability: the RSS is a large matrix and it can easily become non-positive definite, in which case likelihood-based estimation will collapse
- The other reason is computational: We need to fix Q based on some quantity from the data in order to achieve doing only one run of the Kalman Filter
- Based on Kalman filter with forgetting (Jazwinsky, 1970) we can use:

$$Q = \left(\frac{1-\lambda}{\lambda}\right) P_{t-1|t-1} \tag{15}$$

in which case the "predicted" variance of the Kalman Filter becomes

$$P_{t|t-1} = P_{t-1|t-1} + Q = \frac{1}{\lambda} P_{t-1|t-1} \tag{16}$$

$$P_{t|t-1} = P_{t-1|t-1} + Q = \frac{1}{\lambda} P_{t-1|t-1} \quad (17)$$

- The scalar $0 < \lambda \leq 1$ is called a "forgetting factor", forgetting past data at an exponential rate
- For quarterly macroeconomic data, $\lambda = 0.99$ implies observations five years ago receive approximately 80% as much weight as last period's observation
- Put differently, the effective amount of data used for estimation (effective window) is $h = 1/(1-\lambda)$
- With $\lambda = 0.99$ it is the case that $h = 100$ observations are used for estimation
- With $\lambda = 0.5$ it is the case that $h = 2$ observations are used for estimation (can be quite unstable)
- With $\lambda = 1$ we have constant parameters. In this case $h =$
- In practice we set $0.94 < \lambda \leq 1$ or choose λ optimally

- We have taken care of Q, now deal with estimation of Σ_t
- Multivariate stochastic volatility is hard, instead use Exponentially Weighted Moving Average (EWMA)

$$\Sigma_t = \kappa \Sigma_{t-1} + (1-\kappa)\varepsilon_t' \varepsilon_t \tag{18}$$

where ε_t are residuals from the TVP-VAR, i.e. $\varepsilon_t = y_t - z_t \beta_t$ using some estimate of β_t

- The formula is a weighted average of past Σ_t and the Squared Residuals only at time t (and not their sum for t=1 to T)
- $0 < \kappa \leq 1$ is a decay factor, similar to forgetting factor λ
- Effective widow is $w = \frac{\kappa}{2} - 1$
- Again, $\kappa = 1$ gives constant variance
- In practice we set $0.94 < \kappa \leq 1$ or choose κ optimally

- Note: previous slide said we needed some estimate of β_t
- $\beta_{t|t-1}$ from Kalman filter
- This estimate using information through time t-1 to estimate time t quantity (valid for forecasting)
- Thus

$$\varepsilon_t = y_t - z_t \beta_{t|t-1} \tag{19}$$

- So at time t we can update Σ_t and obtain $S_t = (\Sigma_t + z_t P_{t|t-1} z_t')^{-1}$

$$
\begin{aligned}
y_t &= z_t\beta_t + \varepsilon_t \text{ , } \varepsilon_t \sim N(0, \Sigma_t) && (20) \\
\beta_t &= \beta_{t-1} + \eta_t \text{ , } \eta_t \sim N(0, Q) && (21)
\end{aligned}
$$

Fast Kalman filter estimation with variance discounting:

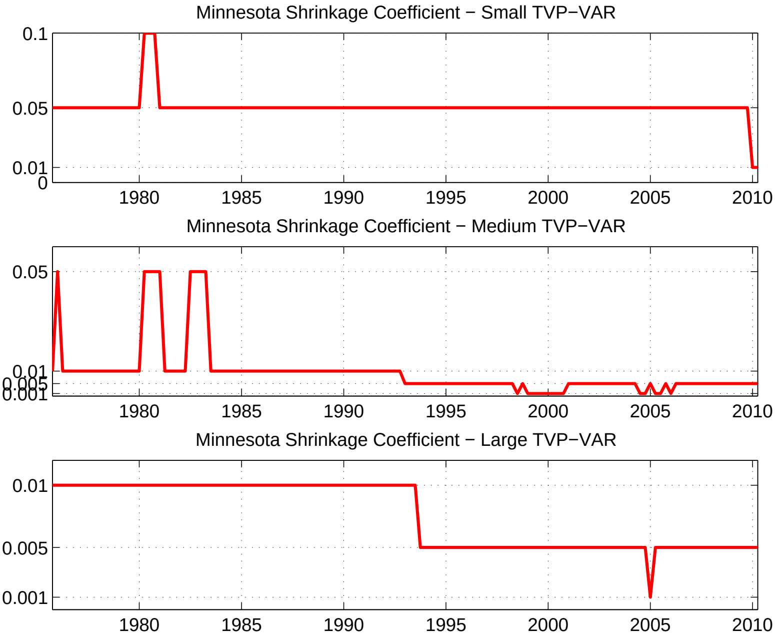
Predict step:

$$
\begin{aligned}
\beta_{t|t-1} &= \beta_{t-1|t-1} && (22) \\
P_{t|t-1} &= \frac{1}{\lambda} P_{t-1|t-1} && (23)
\end{aligned}
$$

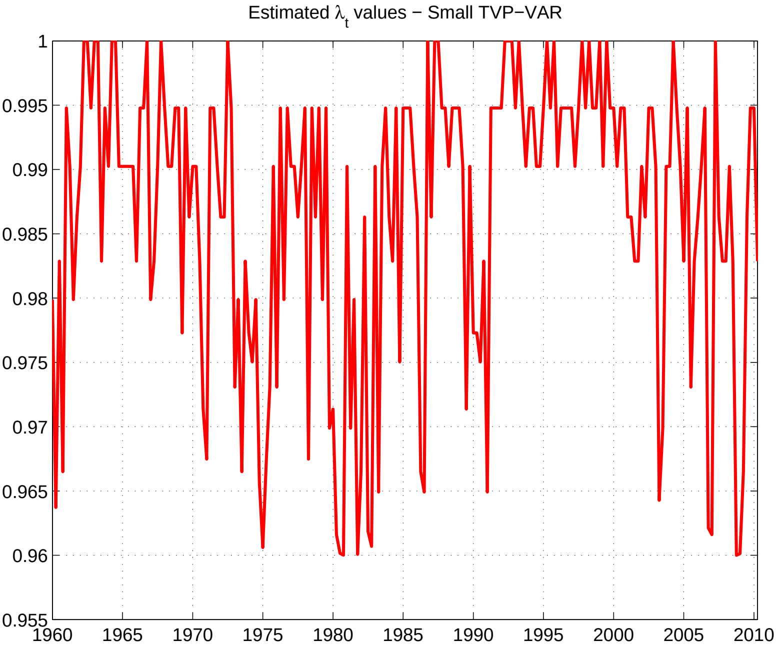
Update step:

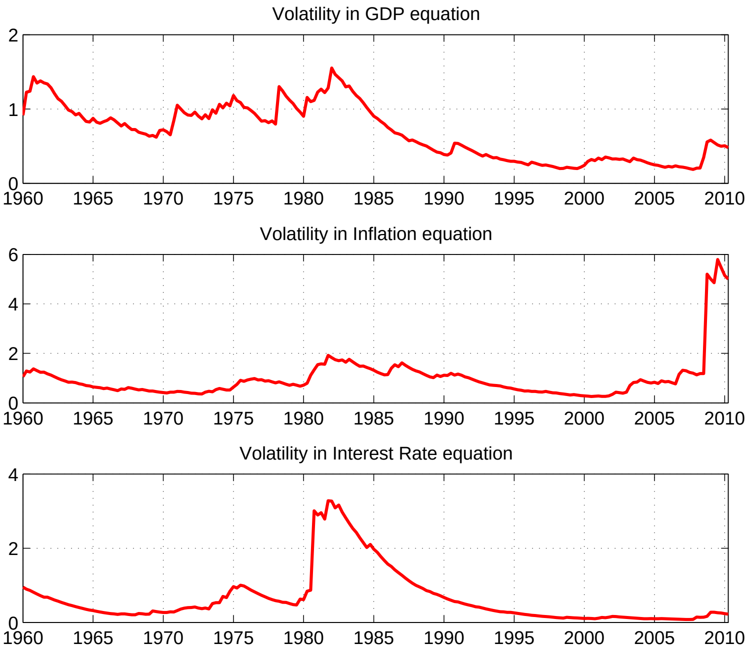
$$
\begin{aligned}
v_{t|t-1} &= y_t - z_t\beta_{t|t-1} && (24) \\
\tilde{\Sigma}_t &= \kappa\tilde{\Sigma}_{t-1} + (1-\kappa) v'_{t|t-1} v_{t|t-1} && (25) \\
S_t &= (\tilde{\Sigma}_t + z_t P_{t|t-1} z'_t)^{-1} && (26) \\
K_t &= P_{t|t-1} z'_t S_t && (27) \\
\beta_{t|t} &= \beta_{t|t-1} + K_t v_{t|t-1} && (28) \\
P_{t|t} &= P_{t|t-1} + K_t z_t P'_{t|t-1} && (29)
\end{aligned}
$$

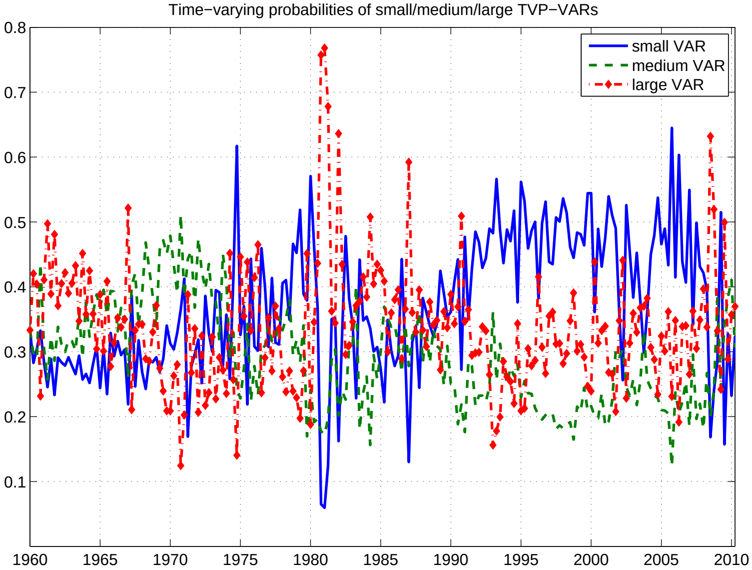
- The algorithm above only involves multiplications and additions of large matrices
- Thus, we can estimate very large systems using this approach
- CPU is not an issue, memory becomes an issue (but no need to save all matrices for all periods t)
- Preceding discussion was all about one TVP-VAR
- Koop and Korobilis (2013) actually have many models
- These models differ in:
- Degree of shrinkage in Minnesota prior for β_0
- Choice of λ and κ
- VAR size having SMALL, MEDIUM, LARGE and large TVP-VARs
- They use dynamic model average (DMA) or selection (DMS) methods
- No time to explain in detail. But these also use forgetting factors.
- Computationally fast even in huge TVP-VARs



Minnesota Shrinkage Coefficient − Small TVP−VAR
0.1
0.05
0.01
0
1980
1985
1990
1995
2000
2005
2010
Minnesota Shrinkage Coefficient − Medium TVP−VAR
0.05
0.01
0.005
0.001
1980
1985
1990
1995
2000
2005
2010
Minnesota Shrinkage Coefficient − Large TVP−VAR
0.01
0.005
0.001
1980
1985
1990
1995
2000
2005
2010


Estimated λt values − Small TVP−VAR
1
0.995
0.99
0.985
0.98
0.975
0.97
0.965
0.96
0.955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010


Volatility in GDP equation
2
1
0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
Volatility in Inflation equation
6
4
2
0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
Volatility in Interest Rate equation
4
2
0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010



Time−varying probabilities of small/medium/large TVP−VARs
small VAR
medium VAR
large VAR
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010

TVP-FAVAR

- FAVAR was discussed in last lecture
- Suitable for Big Data since data are compressed into small number of factors
- Extending to TVP case is difficult due to computational burden
- But fast variance discounting methods can be used to estimate a TVP - FAVAR
- But before I do this, let me go back to the constant parameter case and look again at its estimation
- Even in the constant parameter case estimation is demanding

TVP-FAVAR

- Let x_t be a large number of variables and y_t be a small number of variables chosed as the variables of interest (e.g. the interest rate in the Bernanke, Boivin and Eliasz paper)
- The constant parameter FAVAR can be written as

$$\begin{aligned} \begin{bmatrix} x_t \\ y_t \end{bmatrix} &= \begin{bmatrix} \Lambda & B \\ 0 & I \end{bmatrix} \begin{bmatrix} f_t \\ y_t \end{bmatrix} + \begin{bmatrix} \varepsilon_t \\ 0 \end{bmatrix} \\ \begin{bmatrix} f_t \\ y_t \end{bmatrix} &= \Phi \begin{bmatrix} f_{t-1} \\ y_{t-1} \end{bmatrix} + u_t \end{aligned}$$

- The Dynamic Factor Model (DFM) arises as special case of the FAVAR with no y_t:

$$\begin{aligned} x_t &= \Lambda f_t + \varepsilon_t \\ f_t &= \Phi f_{t-1} + u_t \end{aligned}$$

DFM

- Bayesian can estimate either DFM or FAVAR using MCMC, but computation is demanding
- And they require identifying assumptions
- Principal components (PC) easy
- No identification assumption required (PC implicitly has them)
- Can show PC is good asymptotically.
- In practice, PC good to estimate “static factors” but no dynamics in them
- That is, PC does not use the information in the DFM that $f_t = \Phi f_{t-1} + u_t$

A 2 Step Estimator for the DFM

- Giannone, Reichlin and Sala (2005) suggest an approximate two-step estimation procedure in order to estimate dynamic factors
- Their algorithm is very simple:
  1. Extract PC estimate of f_t, use OLS to estimate all parameters conditional of $\hat{f}_t^{PC}$
  2. Given OLS estimates of parameters run the Kalman filter to estimate f_t
- This approach allows to generate factors which are dynamic (they come from the Kalman filter)
- The correct likelihood-based approach would be to update parameters and factors at the same time (one-step)
- The fact that factors are updated conditional on OLS estimates which are conditional on PC, means that their factor is asymptotically equivalent to PC

A 2 Step Estimator for the DFM

- Doz, Giannone, Reichlin (2011) study this two step approach in more detail
- More precise estimates can be obtained if steps 1-2 in previous slide are iterated many times (EM algorithm)
- Even in this case, though, the final factor is asymptotically equivalent to PC
- In fact with more than 50 series the estimated factor is identical to PC
- There are some differences for 5-20 series (when we extract up to 3 factors)
- In any case, their approach is very useful. It opens new avenues for research.
- Koop and Korobilis (2014, EER) extend this idea in order to estimate a TVP-DFM or TVP-FAVAR

TVP-FAVAR

- We now want to estimate a TVP-FAVAR

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \Lambda_t & B_t \\ 0 & I \end{bmatrix} \begin{bmatrix} f_t \\ y_t \end{bmatrix} + \begin{bmatrix} \varepsilon_t \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} f_t \\ y_t \end{bmatrix} = \Phi_t \begin{bmatrix} f_{t-1} \\ y_{t-1} \end{bmatrix} + u_t$$

where $\varepsilon_t \sim N(0, \Sigma_t)$ and $u_t \sim N(0, \Omega_t)$ and

$$\gamma_t = \gamma_{t-1} + v_t$$

where $\gamma_t = vec(\Lambda_t, B_t, \Phi_t)$

TVP-FAVAR

- We extend the two-step algorithm to allow for TVP as follows:
  1. Estimate $\hat{f}_t^{PC}$ and obtain all estimates of time-varying parameters $(\gamma_t, \Sigma_t, \Omega_t)$
  2. Given the time-varying parameters, update f_t using the Kalman filter/smoother
- For details about how both of these steps work see our paper, ideas on next slide

TVP-FAVAR

- We use the same discounting/forgetting factor approaches used in the large TVP-VAR paper
- Remember these replaced the need for MCMC methods by using simple estimates of key covariances
- Then only needed one run of the Kalman filter/smoother
- Therefore, the algorithm above allows to estimate a TVP-TVP-FAVAR in a fraction of a second - ideal for forecasting
- Alternative is MCMC $\rightarrow$ Large number of latent states, need identification restrictions on Λ_t and i some cases hard to get convergence
- In the two step approach we don't need restrictions on Λ_t, because this parameter is updated conditional on PC

TVP-FAVAR

- Koop and Korobilis (2014) use the TVP-FAVAR to estimate a financial conditions index (FCI)
- Motivation is that different financial variables might be relevant for constructing the FCI, so loadings Λ_t might be time-varying
- E.g. housing market related variables might be relevant during the global financial crisis (higher loadings for that period), but government debt variables might be more important after the crisis
- Additionally, some variables might be completely irrelevant for the FCI in some periods
- Thus we use Dynamic Model Averaging ideas $\rightarrow$ estimate several models with different possible financial variables in x_t
- I have put Malltab code for this model in Computer Session 5.

Summary

- Macroeconomists want to use Big Data and VARs
- Macroeconomists also often need to allow for TVPs or stochastic volatility
- This lecture gone through 3 methods for treating these issues (based on my recent research)
- Code for these models is available on Dimitris Korobilis' website
- https://sites.google.com/site/dimitriskorobilis/matlab
- This is a hot research area now, many new interesting methods coming out each month (and many are Bayesian)