Macroeconomic forecasting using message passing algorithms

Dimitris Korobilis[a]

[a] University of Essex

Norges Bank

Big Data, Machine Learning, and the Macroeconomy

2-3 October 2017

Motivation

- In light of availability of high-dimensional data, machine learning (ML) methods are becoming popular in econometrics
- Athey and Imbens (2017) and Mullainathan and Spiess (2017), both on the same issue of J. of Econ.Persp., highlight the use of such methods in treatment effects, panel data etc.
- Main message: while "traditional" econometrics is all about *consistency*, the ML revolution is mainly about *prediction*
- Recent work by Athey - Imbens showing that such methods can be used for causality and policy evaluation
- Little work has been done in time-series, even though there is recent interest in large VARs and regressions with many predictors

What I do in this paper

- This paper works with the general class of graphical models, estimated using an approximate inference algorithm

What I do in this paper

- This paper works with the general class of graphical models, estimated using an approximate inference algorithm
- First developed by David Donoho (Stanford), and subsequently Sundeep Rangan (NYU), Generalised Approximate Message Passsing (GAMP) has been very successful in signal processing

What I do in this paper

- This paper works with the general class of graphical models, estimated using an approximate inference algorithm
- First developed by David Donoho (Stanford), and subsequently Sundeep Rangan (NYU), Generalised Approximate Message Passsing (GAMP) has been very successful in signal processing
- But little is known about the usefulness of such algorithms in statistics and even less in economics

What I do in this paper

- This paper works with the general class of graphical models, estimated using an approximate inference algorithm
- First developed by David Donoho (Stanford), and subsequently Sundeep Rangan (NYU), Generalised Approximate Message Passsing (GAMP) has been very successful in signal processing
- But little is known about the usefulness of such algorithms in statistics and even less in economics
- Recent attempt by Mike Wand (2017, JASA) to introduce message passing models in statistics (but estimated with Mean Field Variational Bayes methods, not GAMP)

What I do in this paper

- This paper works with the general class of graphical models, estimated using an approximate inference algorithm
- First developed by David Donoho (Stanford), and subsequently Sundeep Rangan (NYU), Generalised Approximate Message Passsing (GAMP) has been very successful in signal processing
- But little is known about the usefulness of such algorithms in statistics and even less in economics
- Recent attempt by Mike Wand (2017, JASA) to introduce message passing models in statistics (but estimated with Mean Field Variational Bayes methods, not GAMP)
- Why should econometricians care about such algorithms at all?

What I do in this paper 2

In this talk I will attempt to establish the following

- GAMP can be easily combined with hierarchical priors that shrink or select/average coefficients (e.g. Normal-iGamma, or spike and slab) and more complex prior structures

What I do in this paper 2

In this talk I will attempt to establish the following

- GAMP can be easily combined with hierarchical priors that shrink or select/average coefficients (e.g. Normal-iGamma, or spike and slab) and more complex prior structures
- The computational cost of GAMP increases linearly in the number of coefficients - it is extremely fast

What I do in this paper 2

In this talk I will attempt to establish the following

- GAMP can be easily combined with hierarchical priors that shrink or select/average coefficients (e.g. Normal-iGamma, or spike and slab) and more complex prior structures
- The computational cost of GAMP increases linearly in the number of coefficients - it is extremely fast
- Unlike "well-established" MCMC algorithms, GAMP-based algorithms require minimal or no tuning. They are simple and more transparent than simulation-based equivalents (i.e. you don't have to be an expert in Bayesian analysis to do variable selection in high-dimensional spaces)

What I do in this paper 2

In this talk I will attempt to establish the following

- GAMP can be easily combined with hierarchical priors that shrink or select/average coefficients (e.g. Normal-iGamma, or spike and slab) and more complex prior structures
- The computational cost of GAMP increases linearly in the number of coefficients - it is extremely fast
- Unlike "well-established" MCMC algorithms, GAMP-based algorithms require minimal or no tuning. They are simple and more transparent than simulation-based equivalents (i.e. you don't have to be an expert in Bayesian analysis to do variable selection in high-dimensional spaces)
- They can be fully modular, trivially parallelizable, and adapted to a wide-class of models

Structure of this talk

This unconvential Introduction was only to convince you that is worth discussing about a methodology that is new to economists and entails lots of engineering jargon. The remainder of the talk is as follows

1. Message passing methods, the GAMP algorithm, and its combination with shrinkage-inducing priors
2. Monte Carlo evaluation
3. Macro Application 1: Forecasting with many orthogonal predictors
4. Macro Application 2: Forecasting inflation with many predictors and time-varying parameters

Methodology

Consider the following regression with the usual notation/assumptions

$$y = x\beta + \varepsilon, \tag{1}$$

where interest lies in estimation of the p-dimensional vector β, with possibly $p \gg T$.

Consider i.i.d prior $p(\beta) = \prod_{i=1}^{p} p(\beta_i)$, then the exact marginal posterior for β_i, $i = 1, ..., p$ requires evaluation of a $(p-1)$-dimensional integral of the form

$$p(\beta_i|y) = \int p(\beta|y)\, d\beta_{j\neq i}, \tag{2}$$

$$\propto \int p(y|\beta)\, p(\beta)\, d\beta_{j\neq i}, \tag{3}$$

$$= p(\beta_i) \int p(y|\beta) \prod_{j=1, j\neq i}^{p} p(\beta_j)\, d\beta_{j\neq i}. \tag{4}$$

Message Passing and Belief Propagation

- I am going now to show that I can approximate this problem using *graphical models*, and in particular *factor graphs*
- Factor graph $\rightarrow$ "factorize (decompose) random variables into quantities of lower dimensions"
- In our case, the high-dimensional random variable we want to factorize is the vector β
  - Example of an efficient factorization: $a \cdot b + a \cdot c = a(b + c)$.
- The following figure depicts the factor graph for the regression problem
  - Variable vertices $\beta = (\beta_1, ..., \beta_p)$ are denoted with a white circle
  - Function vertices $f = [p(\beta), p(y|\beta)] = [p(\beta_1), ..., p(\beta_p), p(y_1|\beta), ..., p(y_T|\beta)]$ are represented using filled boxes

Factor graph for regression model

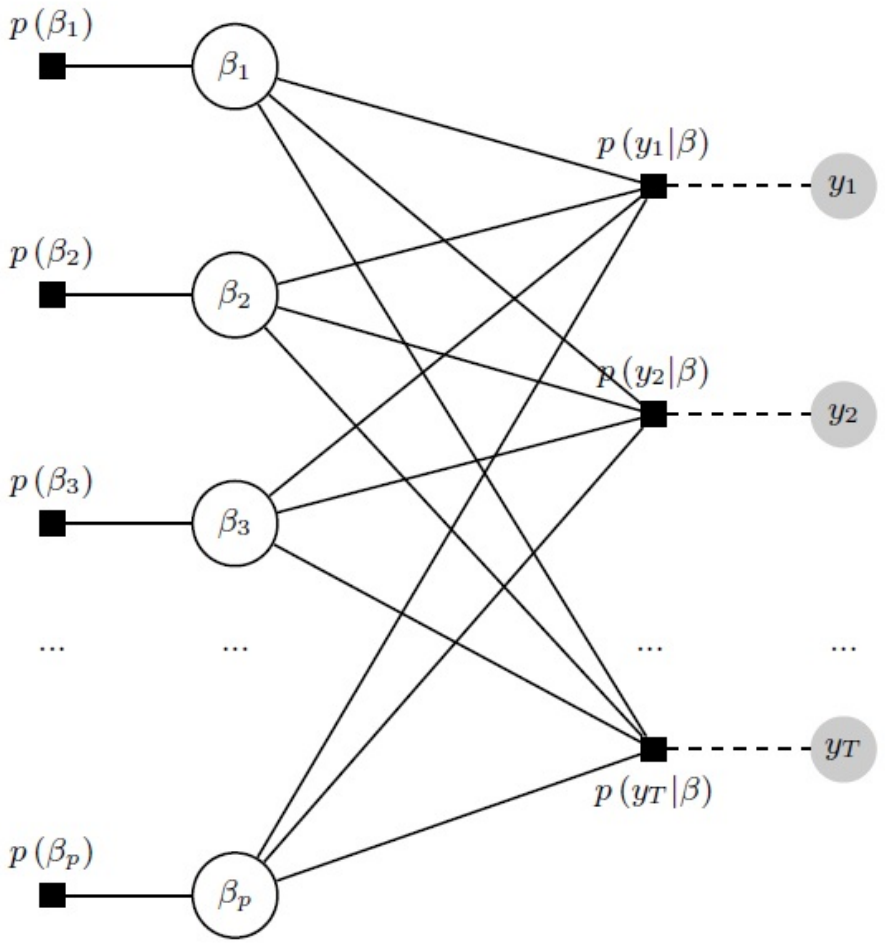


Figure 1: Factor graph for the posterior distribution of β

Generalized Approximate Message Passing

(8)

Message passing approach

Define $\mu_{p(\bullet)\to a}$ the message passed from probability function $p(\bullet)$ to random variable a, then:

$$p\left(\beta_i|y\right) = \mu_{p(\beta_i)\to\beta_i} \prod_{t=1}^{T} \mu_{p(y_t|\beta)\to\beta_i}. \tag{5}$$

where $\mu_{p(\beta_i)\to\beta_i} = p\left(\beta_i\right)$. According to *sum-product* rule we further have

1. $\mu_{p(y_t|\beta)\to\beta_i} = \int p\left(y_t|\beta\right) \prod_{j=1,j\neq i}^{p} \mu_{\beta_j\to p(y_t|\beta)} d\beta_{j\neq i}.$
2. $\mu_{\beta_j\to p(y_t|\beta)} = p\left(\beta_j\right) \prod_{s=1,s\neq t}^{T} \mu_{p(y_s|\beta)\to\beta_j}.$

Belief Propagation

We can estimate messages 1 and 2 above using the following iterative scheme, for $r = 1, ..., R$

$$\mu^{(r+1)}_{p(y_t|\beta)\to\beta_i} = \int p\left(y_t|\beta\right) \prod_{j=1,j\neq i}^{p} \mu^{(r)}_{\beta_j\to p(y_t|\beta)} d\beta_{j\neq i}, \tag{6}$$

$$\mu^{(r+1)}_{\beta_j\to p(y_t|\beta)} = p\left(\beta_j\right) \prod_{s=1,s\neq t}^{T} \mu^{(r)}_{p(y_s|\beta)\to\beta_j}, \tag{7}$$

- This "rule" is called Belief Propagation (Pearl, 1982)
- General approach to inference in Bayesian Networks
- Shows how to calculate the marginal distribution for each unobserved node, conditional on any observed nodes

The GAMP approximation to Belief Propagation

- There are several message passing algorithms for estimating the quantities in the messages $\mu_{p(y_t|\beta)\to\beta_i}$ and $\mu_{\beta_j\to p(y_t|\beta)}$
- For example, Wand (2017, JASA) suggests Variational Bayes
- GAMP relies on two approximations
  - When $p \to \infty$ a central limit theorem (CLT) postulates that the messages $\ln\left[\prod_{j=1,j\neq i}^{p} \mu_{\beta_j\to p(y_t|\beta)}\right]$ can be approximated by a Gaussian distribution with respect to the uniform norm
  - second approximation involves taking the Taylor-series expansion of terms in the messages, so that the mean and variance of $p\left(\beta_i|y\right)$ can be obtained analytically up to the omission of $O\left(1/p\right)$ terms

♠ As $p \to \infty$ both approximations vanish!

The GAMP algorithm in a regression problem

Consider the regression model

$$y_t = x_t\beta + \varepsilon_t,$$

where $\varepsilon_t \sim N\left(0, \sigma^2\right)$, y_t is scalar, x_t is $1 \times p$ vector, and consider the prior distribution $\beta \sim N_p\left(0, \underline{V}\right)$, V diagonal (hence $p(\beta_i)$ independent).

Initialize $\mu_\beta^{(1)} = 0$, $V_\beta^{(1)} = \underline{V}$, set $s^{(0)} = 0$ and define $\widetilde{x} = x \odot x$, where $\odot$ denotes the Hadamard (element-wise) product (also, $\oslash$ denotes element-wise division of matrices/vectors of the same dimension). Finally, define $z = x\beta$.

The GAMP algorithm in practice

Algorithm 1 GAMP iterations

for $r = 1$ **to** R **do**

STEP 1: $\tau_c^{(r)} = (x \odot x)\tau_\beta^{(r)}$

$c^{(r)} = x\mu_\beta^{(r)} - s^{(r-1)} \odot \tau_c^{(r)}$

STEP 2: $s^{(r)} = \left(\mu_z^{(r)} - c^{(r)}\right) \oslash \tau_c^{(r)}$

$\tau_s^{(r)} = \left(1 - \tau_z^{(r)} \oslash \tau_c^{(r)}\right) \oslash \tau_c^{(r)}$

where $\mu_z^{(r)} = E\left(z|\ c^{(r)}, \tau_c^{(r)}\right)$ and $\tau_z^{(r)} = var\left(z|\ c^{(r)}, \tau_c^{(r)}\right)$

STEP 3: $\tau_q^{(r)} = 1 \oslash \left(S\tau_s^{(r)}\right)$

$q^{(r)} = \mu_\beta^{(r)} + \tau_q^{(r)} \odot x's^{(r)}$

STEP 4: $\beta \sim N\left(\mu_\beta^{(r+1)}, \tau_\beta^{(r+1)}\right)$

where $\mu_\beta^{(r+1)} = E\left(\beta|q^{(r)}, \tau_q^{(r)}\right)$ and $\tau_\beta^{(r+1)} = var\left(\beta|q^{(r)}, \tau_q^{(r)}\right)$

end for

Computational points

- The algorithm provides approximation of the first two moments of β (no need to sample from $N\left(\mu_\beta^{(r+1)}, \tau_\beta^{(r+1)}\right)$)
  - $\rightarrow$ Convergence when $\|\mu_\beta^{(r+1)} - \mu_\beta^{(r)}\| \rightarrow 0$
- Total of $\mathcal{O}(Tp)$ algorithmic operations, takes seconds with e.g. $T = 200$ and $p = 200$.
  - $\rightarrow$ Instead of element-wise operators, we can define the algorithm using *for* loops (over t and over i) $\rightarrow$ Easy to parallelize for very large T or p
- Marginalizations/factorizations imply posterior independence, therefore algorithm will not converge with highly correlated predictors.
  - $\rightarrow$ This is a problem for MCMC algorithms (Madigan et al. 1999), but much more so for GAMP.
- Empirically GAMP can still be very useful: will show one Monte Carlo simulation & two examples from macroeconomics

How I use GAMP in this paper

- Main benefit of GAMP: Can be used with generic prior and likelihood functions
- I combine with popular hierarchical priors (using an EM-GAMP scheme):
  1. *Sparse Bayesian Learning* ***(SBL)***, (Tipping, 2001, J.of Machine Learning Research):

$$p(\beta_i|\alpha_i) = N\left(0, \alpha_i^{-1}\right), \quad (8)$$

$$p(\alpha_i) = Gamma(\underline{a}, \underline{b}). \quad (9)$$

  2. *Spike and Slab prior* ***(SNS)*** (Mitchell and Beauchamp, 1988, JASA):

$$p(\beta_i|\pi_0) = (1-\pi_0)\delta_0 + \pi_0 N\left(0, \alpha^{-1}\right), \quad (10)$$

$$p(\pi_0) = Beta\left(\underline{\rho}_1, \underline{\rho}_2\right). \quad (11)$$

Experiments on artificial data

- Generate p predictors, with T observations each, from a Normal
- Correlation among predictors is $corr(x_i, x_j) = \rho^{|i-j|}$
- Only first $q = \lfloor c \times p \rceil$ predictors important
- q coefficients from continuous $U(-4, 4)$, all others are zero

Three different Data Generating Processes

1. Model 1: $T = 50$, $p = 100, 200, 500$ and $\rho = 0.3$ and $c = 0.01$
2. Model 2: $T = 200$, $p = 100, 200, 500$ and $\rho = 0.3$ and $c = 0.05$
3. Model 3: $T = 200$, $p = 100$ and $\rho = 0.9$ and $c = 0.05$.
   $\rightarrow$ In Model 3 predictors are orthogonalized (for estimation)

♦ Measure accuracy using Mean Absolute Deviations between true and estimated parameters over all Monte Carlo iterations

Model 1: Boxplots of MAD statistics

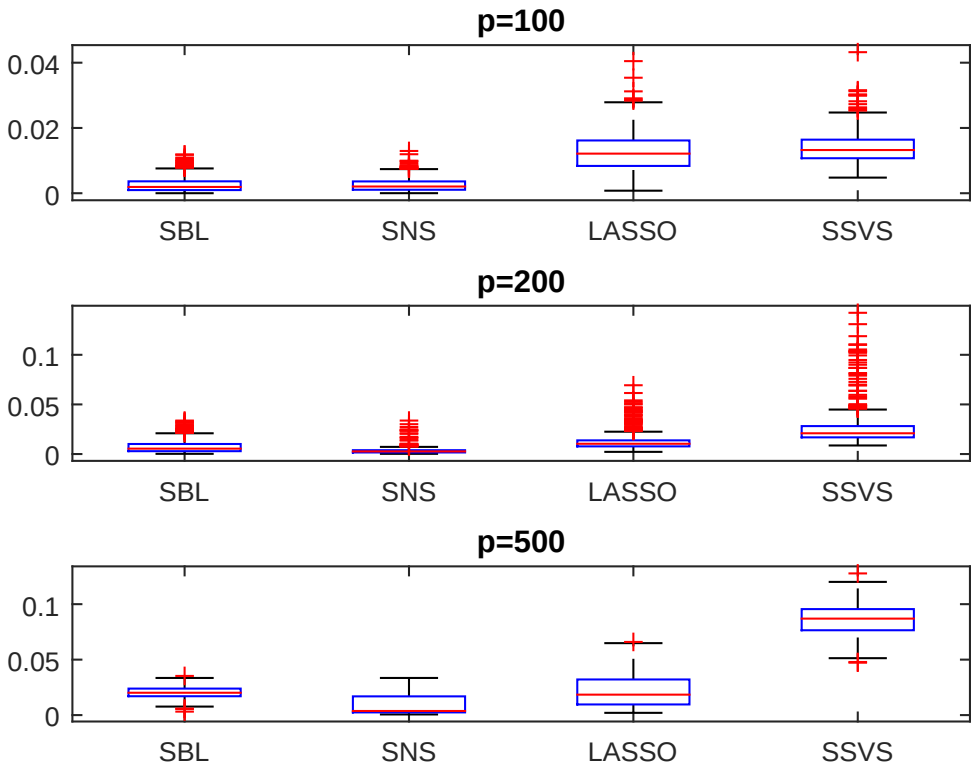


Figure: Boxplots of MAD statistics over the 500 Monte Carlo iterations for Model 1 case ($T = 50$, $p = 100, 200, 500$).

Model 2: Boxplots of MAD statistics

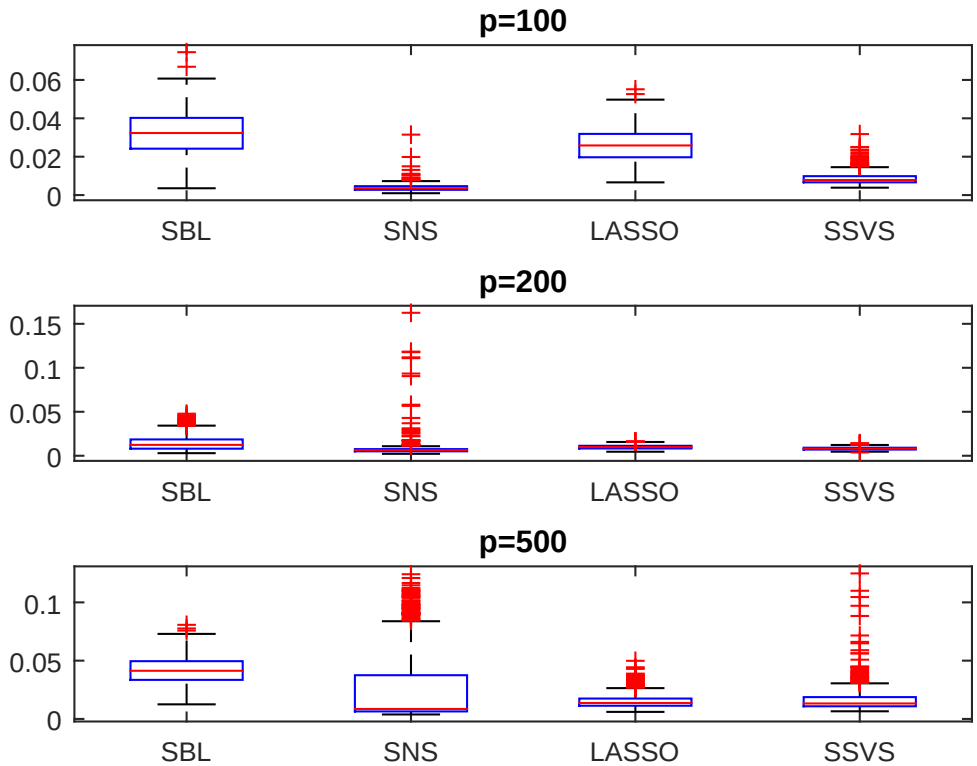


Figure: Boxplots of MAD statistics over the 500 Monte Carlo iterations for Model 2 case ($T = 200$, $p = 100, 200, 500$).

Model 3: Boxplots of MAD statistics

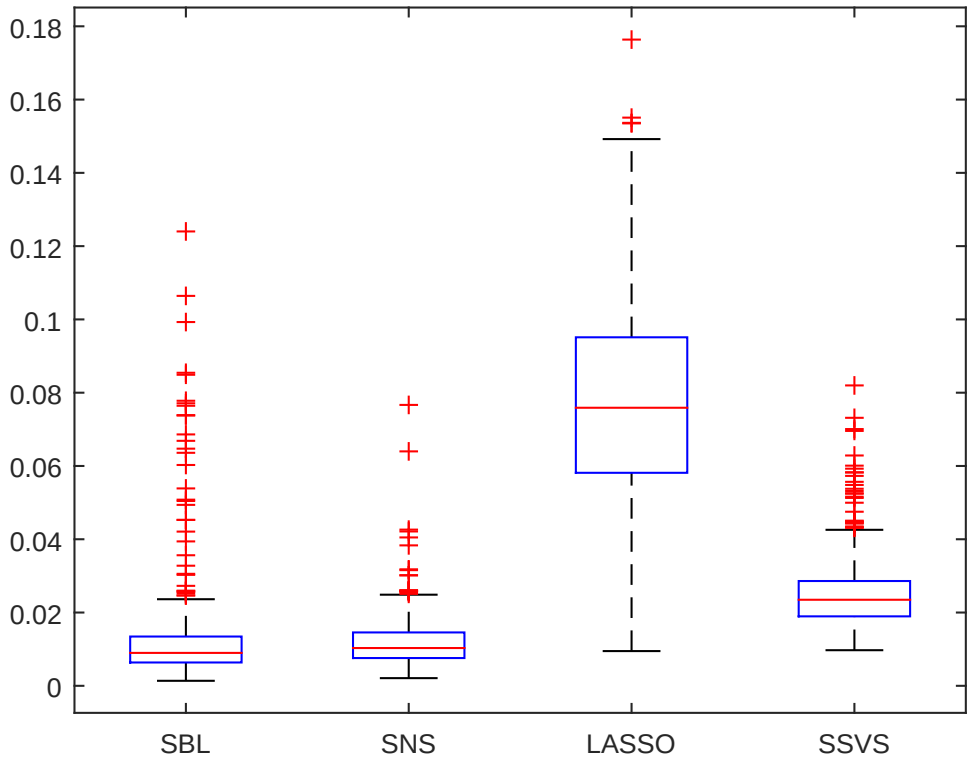


Figure: Boxplots of MAD statistics over the 500 Monte Carlo iterations for Model 3 case ($T = 200$, $p = 100$).

...and why you should take this algorithm seriously

Table: Computing time (seconds) per Monte Carlo iteration for each of the four algorithms.

		SBL	SNS	LASSO	SSVS
$T = 50$	$p = 100$	< 0.01	< 0.01	4.03	1.81
	$p = 200$	< 0.01	< 0.01	12.88	5.29
	$p = 500$	< 0.01	0.01	92.99	38.59
$T = 200$	$p = 100$	< 0.01	0.01	6.28	1.98
	$p = 200$	< 0.01	0.01	12.54	5.11
	$p = 500$	< 0.01	0.28	54.89	18.29

Notes: The reference machine is a 64 bit Windows 7-based PC with Intel Core 7 4770K CPU, 32GB DDR3 RAM running MATLAB 2016a.

Empirical Application 1

Now let's look how this algorithm fairs in real situations

- 222 US quarterly macro series, 1959:Q1-2015:Q3 (FRED-QD of Mike McCracken)
- I use one series at a time as y, remaining 221 converted to factors (to be exact, only 130 of 222 disaggregated series used for factors)
- Standard univariate forecast regressions with one lag + 50 factors (orthogonal predictors)
- $h = 1, 2, 4, 8$ (setting identical to Stock and Watson, 2012, JBES)
- Competing methods: LASSO, SSVS, BMA, BAGGING, DFM5 (use always first 5 factors from 50), OLS (on the full model with 50 factors)

Tables show quantiles of MSFEs (relative to MSFE of $AR(1)$) averaged over all 222 series

Empirical Application 1, constant volatility

				Horizon $h = 1$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.746	0.763	0.781	0.723	0.733	0.757	0.745	0.872
0.25	0.898	0.935	0.943	0.917	0.915	0.921	0.915	1.033
0.5	0.975	0.999	0.996	0.990	0.990	0.987	0.993	1.130
0.75	1.011	1.002	1.040	1.003	1.013	1.020	1.029	1.273
0.95	1.058	1.021	1.111	1.037	1.068	1.087	1.106	1.544
				Horizon $h = 2$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.745	0.737	0.799	0.748	0.750	0.713	0.753	0.844
0.25	0.866	0.911	0.919	0.889	0.897	0.902	0.887	1.013
0.5	0.980	0.998	1.001	0.990	0.989	0.979	0.990	1.121
0.75	1.021	1.002	1.050	1.007	1.024	1.033	1.041	1.270
0.95	1.094	1.052	1.135	1.069	1.112	1.114	1.137	1.495
				Horizon $h = 4$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.765	0.758	0.784	0.774	0.766	0.703	0.783	0.858
0.25	0.871	0.888	0.908	0.885	0.884	0.885	0.870	1.014
0.5	0.961	0.979	0.991	0.975	0.980	0.970	0.978	1.111
0.75	1.012	1.001	1.054	1.009	1.030	1.025	1.051	1.279
0.95	1.129	1.057	1.163	1.102	1.166	1.159	1.197	1.574
				Horizon $h = 8$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.728	0.703	0.753	0.696	0.753	0.769	0.744	0.892
0.25	0.909	0.927	0.920	0.916	0.917	0.935	0.898	1.036
0.5	0.989	0.993	1.010	0.985	1.001	1.008	0.973	1.183
0.75	1.047	1.019	1.078	1.028	1.054	1.070	1.060	1.357
0.95	1.157	1.111	1.232	1.136	1.236	1.253	1.321	1.895

Empirical Application 1, EWMA volatility

			Horizon $h = 1$		
	SBL-EWMA	**SNS-EWMA**	**SBL**	**SNS**	**DFM5**
0.05	0.730	0.733	0.746	0.763	0.745
0.25	0.892	0.928	0.898	0.935	0.915
0.5	0.971	0.991	0.975	0.999	0.993
0.75	0.999	0.994	1.011	1.002	1.029
0.95	1.028	1.018	1.058	1.021	1.106
			Horizon $h = 2$		
	SBL-EWMA	**SNS-EWMA**	**SBL**	**SNS**	**DFM5**
0.05	0.747	0.738	0.745	0.737	0.753
0.25	0.858	0.892	0.866	0.911	0.887
0.5	0.972	0.990	0.980	0.998	0.990
0.75	1.002	0.995	1.021	1.002	1.041
0.95	1.056	1.027	1.094	1.052	1.137
			Horizon $h = 4$		
	SBL-EWMA	**SNS-EWMA**	**SBL**	**SNS**	**DFM5**
0.05	0.752	0.747	0.765	0.758	0.783
0.25	0.868	0.882	0.871	0.888	0.870
0.5	0.946	0.979	0.961	0.979	0.978
0.75	0.999	0.994	1.012	1.001	1.051
0.95	1.069	1.051	1.129	1.057	1.197
			Horizon $h = 8$		
	SBL-EWMA	**SNS-EWMA**	**SBL**	**SNS**	**DFM5**
0.05	0.703	0.675	0.728	0.703	0.744
0.25	0.893	0.910	0.909	0.927	0.898
0.5	0.976	0.982	0.989	0.993	0.973
0.75	1.027	1.005	1.047	1.019	1.060
0.95	1.120	1.090	1.157	1.111	1.321

Empirical Application 2

- Second application involves TVP regressions for inflation

$$\pi_t = \tau_t + \phi_t \pi_{t-1} + \beta_t x_t + \varepsilon_t \tag{12}$$

  We now know that for inflation SV is important (as with other macro variables), but also time-varying trend and possibly lags
- Many people have proposed various Bayesian methods - some are in this room today.
- Existing methods cannot incorporate large number of predictors
- Even if they do incorporate some predictors MCMC variants are extremely complex, and hard to beat parsimonious TVP-AR or UC-SV specifications or simpler forgetting factor estimators

Empirical Application 2

The TVP model can be written as constant parameter regression with time-dummies (assume intercept and lags are all in a matrix x)

$$y = z\theta + \varepsilon \tag{13}$$

where

$$z = \begin{bmatrix} x_1 & 0 & . & . & 0 \\ 0 & x_2 & . & . & . \\ . & 0 & . & . & . \\ . & . & . & . & . \\ 0 & 0 & . & . & x_T \end{bmatrix}.$$

- Without further assumptions, $z'z$ is rank deficient
- This is why people rely on the hierarchical prior $p\left(\theta_t|\theta_{t-1}\right) \sim RW$
- What if I do not use this restrictive prior, but instead **shrink** θ?

Empirical Application 2

$$y = z\theta + \varepsilon \tag{14}$$

- What if I do not use this restrictive prior, but instead **shrink** θ?

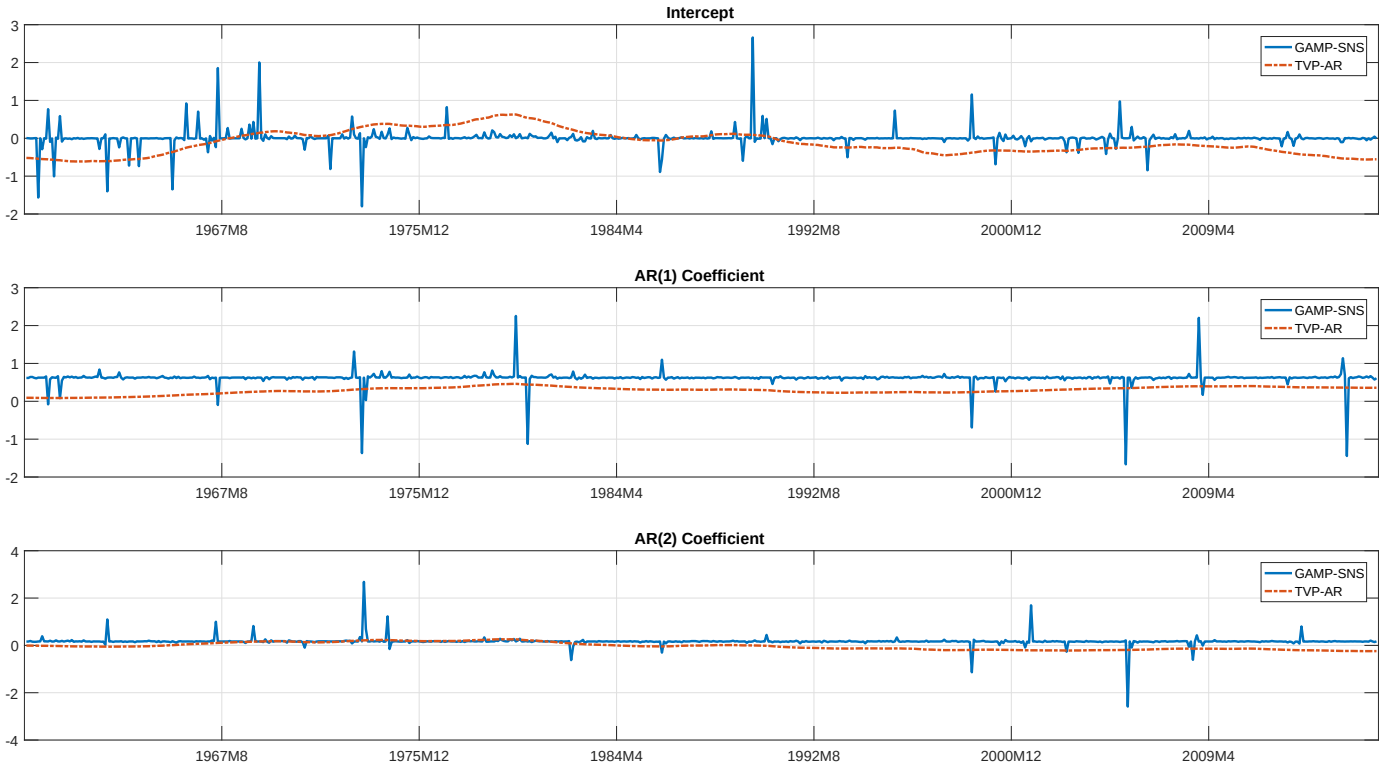
- I will now show that GAMP can be used to estimate this regression and recover TVP coefficients
- I use monthly CPI data (685 obs) and 118 predictors
- z has 80,830 columns!

→ Note that if desirable, we can also do the same shrinkage estimation WITH the RW prior imposed, WITHOUT the need of Kalman Filter (see Chan and Jeliazkov, 2009)

→ In that case, we could also use GAMP in state-space form

Empirical Application 2: Parameter Estimates

Figure: Parameter estimates (Int, first & second lag), TVP-AR (MCMC) vs Regression with time dummies + shrinkage (GAMP)



Empirical Application 2: MSFEs (relative to AR(2))

	$h=1$	$h=6$	$h=12$
AR(2)	1.000	1.000	1.000
TVP AR models:			
KP-AR(2) (Koop+Potter)	0.878	0.610	0.514
GK-AR(2) (Giordani+Kohn)	0.975	0.878	0.839
TVP-AR(2)	0.888	0.808	0.626
UC-SV	0.996	0.944	0.568
Constant parameter models with predictors:			
LASSO-AR(2) + all predictors	0.901	0.863	0.916
SSVS-AR(2) + all predictors	0.898	0.914	0.894
GAMP-SBL-AR(2) + all predictors	0.999	0.822	0.927
GAMP-SNS-AR(2) + all predictors	0.952	0.895	0.740
TVP models with predictors:			
TVP-BMA-AR(2) + 10 predictors	0.906	1.070	0.813
TVP-DMA-AR(2) + 10 predictors	0.825	0.594	0.433
GAMP-SBL-TDAR(2) + all predictors	0.863	0.690	0.505
GAMP-SNS-TDAR(2) + all predictors	0.832	0.701	0.604

Empirical Application 2: logPLs (relative to AR(2))

	$h=1$	$h=6$	$h=12$
AR(2)	0.000	0.000	0.000
TVP AR models:			
KP-AR(2) (Koop+Potter)	-0.089	0.522	0.218
GK-AR(2) (Giordani+Kohn)	0.202	0.078	-0.076
TVP-AR(2)	0.287	0.112	0.096
UC-SV	0.065	0.078	0.101
Constant parameter models with predictors:			
LASSO-AR(2) + all predictors	-0.515	-0.997	-0.333
SSVS-AR(2) + all predictors	0.191	0.489	0.268
GAMP-SBL-AR(2) + all predictors	-0.313	0.123	0.031
GAMP-SNS-AR(2) + all predictors	0.115	0.194	-0.192
TVP models with predictors:			
TVP-BMA-AR(2) + 10 predictors	0.548	0.077	0.551
TVP-DMA-AR(2) + 10 predictors	-0.091	-0.482	-0.178
GAMP-SBL-TDAR(2) + all predictors	0.003	0.416	0.045
GAMP-SNS-TDAR(2) + all predictors	0.055	0.258	0.144

Conclusions

- New Bayesian estimation methodology for models with orthogonal predictors
- Extremely fast, automatic (not much tuning), and low maintenance costs
- Fully modular and parallelizable algorithm
- Can be combined with arbitrary prior distributions - I show how it works with Normal-Inverse Gamma, and spike and slab priors
- It can be useful in various scenarios that macroeconomists are interested in
- But algorithm can be combined with arbitrary likelihood function - possible future research would be to use GAMP to estimate Bayesian quantile regression (Laplace likelihood)