Macroeconomic forecasting using message passing algorithms

Dimitris Korobilis[a]

[a] University of Essex

NBP Workshop on Forecasting

26-27 November 2018

Motivation

- In light of availability of high-dimensional data, machine learning (ML) methods are becoming popular in econometrics
- Athey and Imbens (2017, JEP) and Mullainathan and Spiess (2017, JEP) 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, Chernozukhov etc on proving 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

Contribution:

1. Introduce to the econometrics literature machine learning methodologies for designing efficient Bayesian algorithms

2. Introduce a novel interpretation and treatment of the time-varying parameter regression as a high-dimensional shrinkage problem.

First contribution

1 Introduce to the econometrics literature machine learning methodologies for designing efficient Bayesian algorithms

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

First contribution

1 Introduce to the econometrics literature machine learning methodologies for designing efficient Bayesian algorithms

- 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

First contribution

1 Introduce to the econometrics literature machine learning methodologies for designing efficient Bayesian algorithms

- 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

First contribution

1 Introduce to the econometrics literature machine learning methodologies for designing efficient Bayesian algorithms

- 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 inference in statistics

Second contribution

2 Introduce a novel interpretation and treatment of the time-varying parameter regression as a high-dimensional shrinkage problem.

- At least since the works of Cooley/Prescott, Sims/Sargent etc in the early 70s, time-varying parameter (TVP) models are now the standard

Second contribution

2 Introduce a novel interpretation and treatment of the time-varying parameter regression as a high-dimensional shrinkage problem.

- At least since the works of Cooley/Prescott, Sims/Sargent etc in the early 70s, time-varying parameter (TVP) models are now the standard
- In one of his last papers Granger (2008) was speculating that TVP models will be the future of time-series modeling

Second contribution

2 Introduce a novel interpretation and treatment of the time-varying parameter regression as a high-dimensional shrinkage problem.

- At least since the works of Cooley/Prescott, Sims/Sargent etc in the early 70s, time-varying parameter (TVP) models are now the standard
- In one of his last papers Granger (2008) was speculating that TVP models will be the future of time-series modeling
- With few recent exceptions (e.g. works by Kapetanios /Giraitis) TVP models rely solely on state-space representation

Second contribution

2 Introduce a novel interpretation and treatment of the time-varying parameter regression as a high-dimensional shrinkage problem.

- At least since the works of Cooley/Prescott, Sims/Sargent etc in the early 70s, time-varying parameter (TVP) models are now the standard
- In one of his last papers Granger (2008) was speculating that TVP models will be the future of time-series modeling
- With few recent exceptions (e.g. works by Kapetanios /Giraitis) TVP models rely solely on state-space representation
- Instead treat the TVP estimation problem as a high-dimensional static regression using shrinkage priors – No reliance on Kalman filter

TVP Regression

Starting point:

$$y_t = x_t \beta_t + \varepsilon_t \tag{1}$$

- y_t is the tth observation on the variable of interest, $t = 1, ..., T$
- x_t is a $1 \times p$ vector of predictors (possibly including lags of y_t)
- β_t is a $p \times 1$ vector of coefficients
- $\varepsilon_t \sim N\left(0, \sigma_t^2\right)$ with σ_t^2 the time-varying variance parameter.

Static form of TVP Regression

Equivalent static form

$$y = \mathcal{X}\beta + \varepsilon, \tag{2}$$

- $y = [y_1, ..., y_T]'$ and $\varepsilon = [\varepsilon_1, ..., \varepsilon_T]'$ are column vectors stacking the observations y_t and ε_t respectively
- $\beta = [\beta_1', ..., \beta_T']'$ is a $Tp \times 1$ vector
- $\mathcal{X}$ is the $T \times Tp$ matrix

$$\mathcal{X} = \begin{bmatrix} x_1 & 0_{1\times p} & \cdots & 0_{1\times p} & 0_{1\times p} \\ 0_{1\times p} & x_2 & \cdots & 0_{1\times p} & 0_{1\times p} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0_{1\times p} & 0_{1\times p} & \cdots & x_{T-1} & 0_{1\times p} \\ 0_{1\times p} & 0_{1\times p} & \cdots & 0_{1\times p} & x_T \end{bmatrix}, \tag{3}$$

Static form of TVP Regression

Equivalent static form

$$y = \mathcal{X}\beta + \varepsilon, \tag{4}$$

- Gram matrix $\mathcal{X}'\mathcal{X}$ has only rank $T \rightarrow$ OLS has no unique solution
- Typical approach is to define an AR/RW model for β_t
- For a Bayesian, such model can be thought of as a hierarchical prior of the form $p\left(\beta_t|\beta_{t-1}\right) \sim N\left(\beta_{t-1}, Q\right)$
- Such prior is very informative, and leads to some shrinkage
- However, equation (24) can be estimated using better hierarchical shrinkage priors (LASSO, spike and slab etc)

Sparse Bayesian Learning prior

Static regression form

$$y = \mathcal{X}\beta + \varepsilon, \tag{5}$$

I follow Tipping (2001, JMLR) and use "Sparse Bayesian Learning" prior for each element β_i of the vector β, $i = 1, 2, ..., Tp$,

$$p(\beta_i|\alpha_i) = N\left(0, \alpha_i^{-1}\right), \tag{6}$$

$$p(\alpha_i) = Gamma(\underline{a}, \underline{b}). \tag{7}$$

♣ This conditionally Normal prior for β_i is a scale mixtures of Normal representation of a *Student-t prior*.

Dealing with stochastic volatility 1

- In this static form, β is huge and will be estimated efficiently with message passing
- Of interest is to find out how to estimate $\sigma^2 = \left(\sigma_1^2, ..., \sigma_T^2\right)'$

Define Σ a $T \times T$ diagonal matrix with element σ_t^2, and write

$$y = \mathcal{X}\beta + \Sigma v, \tag{8}$$

Conditional on knowing β

$$\log\left[(y - \mathcal{X}\beta)^2\right] = \log\left(diag\left(\Sigma\right)^2\right) + \log(v^2), \Rightarrow \tag{9}$$

$$\widetilde{y} = \widetilde{\sigma}^2 + \widetilde{v}, \tag{10}$$

♣ If $\widetilde{v}$ was Normal, then problem above is that estimation of a Normal mean ($\widetilde{\sigma}^2$)

Dealing with stochastic volatility 2

- $\widetilde{v}$ is $\log -\chi^2$ and not Normal
- Follow Kim, Shephard, Chib (1998) and approximate using mixture of 7 Normals, with means μ_i, variances V_i, weights π_i
- Estimate

$$\widetilde{y} = \widetilde{\sigma}^2 + u_i, \qquad \text{for } i = 1, ..., 7, \tag{11}$$

where $u_i \sim N(\mu_i, V_i)$

An estimator of the $T \times 1$ vector of log-volatilities is of the form $E_i\left(\widetilde{\sigma}^2\right) = \widetilde{y} - \mu_i$, and the final volatility estimate at time t is

$$\widehat{\sigma}_t^2 = \exp\left(\sum_{i=1}^{7} \pi_i \left(\widetilde{y}_t - \mu_i\right)\right)/7. \tag{12}$$

Introduction to Factor Graphs

Some definitions

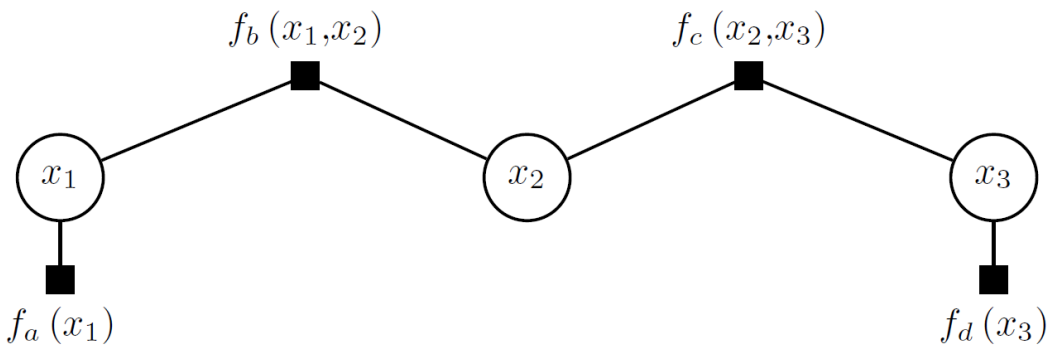
- ***Graphical model:*** *probabilistic structure that relies on graph theory in order to express conditional dependence between random variables*
- ***Factor graph:*** *Bipartite graph that represents the way a global distribution of several random variables is decomposed into a product of simpler functions ("factors").*
- ***Message passing:*** *Dynamic programming solutions, where a node collects a result from a part of the graph and communicates it to the next neighboring node via a message.*
- ***Sum-product algorithm:*** *A rule specifying the way each node collects all messages in order to calculate the marginal distribution of that message.*

A toy example

- Consider discrete random variables $x = (x_1, x_2, x_3)$
- And a joint mass function p that we can decompose, say, as

$$p(x_1, x_2, x_3) = f_a(x_1) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3), \quad (13)$$

  where f_a, f_b, f_c, f_d are the factors that have a certain functional known to us analytically
- Factor graph for this example:



Marginal distribution of x_i

Marginal distribution of a generic x_i is

$$p(x_i) = \sum_{x \setminus x_i} p(x) . \tag{14}$$

which can be computationally cumbersome in high-dimensions and for continuous x

♣ Idea is to calculate the marginal distribution using *message passing* algorithms

- Denote as $\mu_{x_i \to f_j}$ the message sent from variable x_i to function f_j
- $\mu_{f_j \to x_i}$ the message sent from factor node f_j to variable node x_i

Message passing inference

1. Message sent from x_i to f_j: product of all messages arriving to node x_i except from the message coming from the target node f_j:

$$\mu_{x_i \to f_j} = \prod_{k \in N(x_i), k \neq j} \mu_{f_k \to x_i}, \tag{15}$$

2. Message from node f_j to node x_i: sum over the product of the factor function f_j itself and all the incoming messages, except the messages from the target variable node x_i:

$$\mu_{f_j \to x_i} = \sum_{x \setminus x_i} f_j(x) \prod_{l \in N(x_i), l \neq i} \mu_{x_l \to f_j}, \tag{16}$$

3. If x_i is a leaf variable node, $\mu_{x_i \to f_j} = 1$
4. If f_j is a leaf factor, $\mu_{f_j \to x_i} = f_j(x_i)$

Marginal distribution of x_i: $p(x_i) \propto \prod_{m \in N(x_i)} \mu_{f_m \to x_i}$.

Example calculation of $p(x_2)$ in toy example

- Messages emitted to node x_2 from left are

$$\mu_{f_a \to x_1} = f_a(x_1), \tag{17}$$

$$\mu_{x_1 \to f_b} = \mu_{f_a \to x_1} = f_a(x_1), \tag{18}$$

$$\mu_{f_b \to x_2} = \sum_{x_1} f_b(x_1 x_2)\, \mu_{x_1 \to f_b}, \tag{19}$$

- Messages emitted to node x_2 from right are

$$\mu_{f_d, x_3} = f_d(x_3), \tag{20}$$

$$\mu_{x_3 \to f_c} = \mu_{f_c \to x_3} = f_d(x_3), \tag{21}$$

$$\mu_{f_c \to x_2} = \sum_{x_3} f_c(x_2, x_3)\, \mu_{x_3 \to f_c}, \tag{22}$$

The marginal distribution of x_2 is

$$p(x_2) \propto \mu_{f_b \to x_2} \times \mu_{f_c \to x_2}. \tag{23}$$

Regression representation using factor graphs

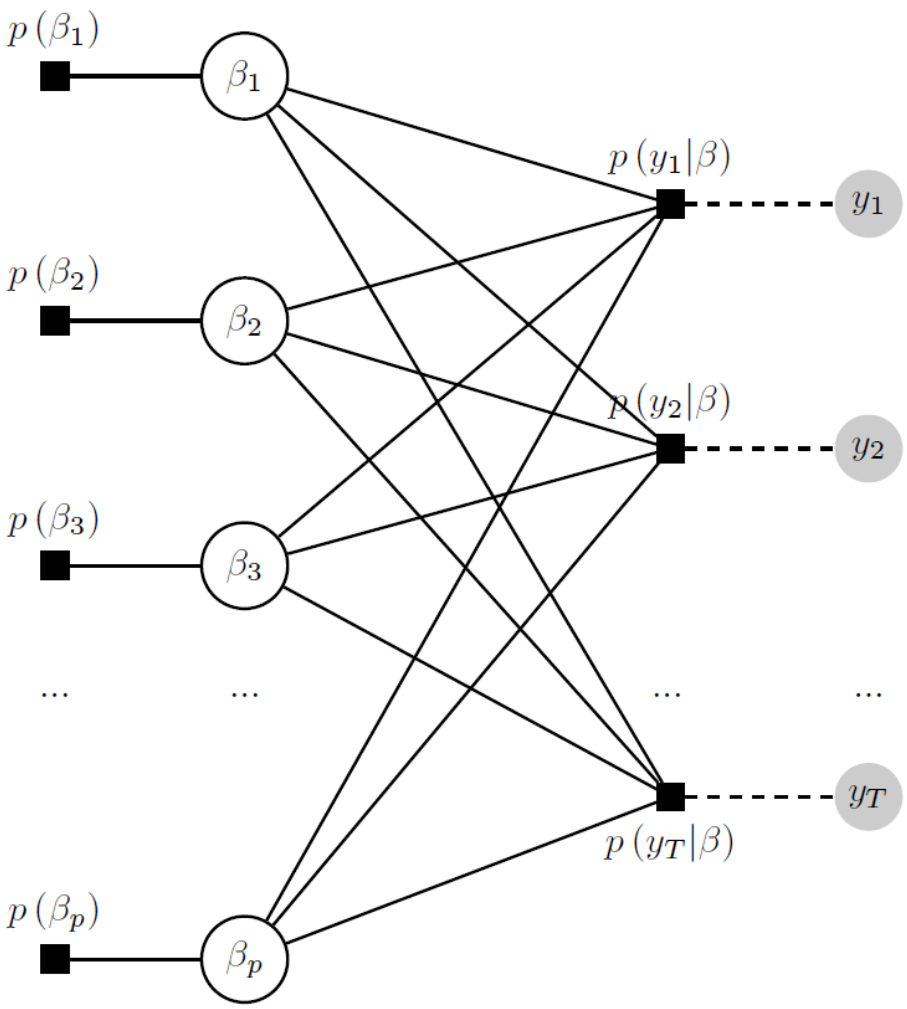
- Let's go back to our regression problem, in its static form this was

$$y = \mathcal{X}\beta + \varepsilon, \tag{24}$$

- For simplicity assume $\sigma_t^2 = \underline{\sigma}$, fixed and known
- Focus on $q \times 1$ vector of random variables β. The exact marginal posterior for β_i, $i = 1, ..., q$ is of the form

$$\begin{aligned} p(\beta_i|y) &= \int p(\beta|y)\, d\beta_{j\neq i}, && (25) \\ &\propto \int p(y|\beta)\, p(\beta)\, d\beta_{j\neq i}, && (26) \\ &= p(\beta_i) \int p(y|\beta) \prod_{j=1, j\neq i}^{q} p(\beta_j)\, d\beta_{j\neq i}, && (27) \end{aligned}$$

Factor graph representation of Bayesian regression



Factor graph representation of Bayesian regression

- Redefine the marginal posterior of β_i as the product of incoming messages at node β_i in the graph

$$p\left(\beta_i|y\right) = \mu_{p(\beta_i)\to\beta_i} \prod_{t=1}^{T} \mu_{p(y_t|\beta)\to\beta_i}. \tag{28}$$

- Message $\mu_{p(\beta_i)\to\beta_i}$ is leaf factor node, hence equal to $p\left(\beta_i\right)$.
- Following the sum-product message rule we can write

$$\mu_{p(y_t|\beta)\to\beta_i} = \int p\left(y_t|\beta\right) \prod_{j=1,j\neq i}^{p} \mu_{\beta_j\to p(y_t|\beta)} d\beta_{j\neq i}. \tag{29}$$

- Message from node β_j to function $p\left(y_t|\beta\right)$ is the product of all incoming messages to node β_i, excluding $p\left(y_t|\beta\right)$

$$\mu_{\beta_j\to p(y_t|\beta)} = p\left(\beta_j\right) \prod_{s=1,s\neq t}^{T} \mu_{p(y_s|\beta)\to\beta_j}. \tag{30}$$

Factor graph representation of Bayesian regression

- If I haven't "lost" you so far, bottom line of all this is that if I iterate through the following

$$\mu^{(r+1)}_{p(y_t|\beta)\to\beta_i} = \int p\left(y_t|\beta\right) \prod_{j=1,j\neq i}^{q} \mu^{(r)}_{\beta_j\to p(y_t|\beta)} d\beta_{j\neq i}, \quad (31)$$

$$\mu^{(r+1)}_{\beta_j\to p(y_t|\beta)} = p\left(\beta_j\right) \prod_{s=1,s\neq t}^{T} \mu^{(r)}_{p(y_s|\beta)\to\beta_j}, \quad (32)$$

after a few iterations I should be able to get the marginal posterior of β_i using equation (28)

♣ Given the "factors" $p(y|\beta)$ and $p(\beta)$, can I derive analytically the functional forms implied by the messages, e.g. $\mu_{\beta_j\to p(y_t|\beta)}$?

GAMP

♣ Given the "factors" $p(y|\beta)$ and $p(\beta)$, can I derive analytically the functional forms implied by the messages, e.g. $\mu_{\beta_j \to p(y_t|\beta)}$?

- Answer is "no"
- This is where Generalized Approximate Message Passing (GAMP) comes to play
- You can think of the iterative sum-product rule as the equivalent to the "Kalman filter (KF)" for state-space models, i.e. a generic solution under certain regularity conditions
- But GAMP can be thought of as one of many algorithmic implementations of the KF when the parameter matrices are not known (e.g. MCMC-KF, ML-KF, EM-KF etc).

What does GAMP do?

- Derivation of GAMP is complex so will explain in few words and refer you to the Appendix of my paper
- GAMP introduces Normal approximations to the messages $\mu_\bullet$ defined above
- First, when $q \to \infty$ a central limit theorem (CLT) postulates that the messages $\prod_{j=1,j\neq i}^{q} \mu_{\beta_j \to p(y_t|\beta)}$ are Gaussian with respect to the uniform norm
- Second, take Taylor series expansion of messages such that mean and variance of $p(\beta_i|y)$ can be approximated up to omission of $O\left(1/q\right)$ terms
- **Both approximations vanish as $q \to \infty$: "Blessing of Big Data", rather than curse of dimensionality**

What does GAMP look like?

Initialize $\widehat{\beta}_j^{(0)} = 0$ and $\widehat{\tau}_j^{\beta,(0)} = 100\ \forall j = 1, \ldots, q$, and set $\widehat{s}_t^{(0)} = 0\ \forall t = 1, \ldots, T$.
r=1
while $\|\widehat{\beta}^{(r)} - \widehat{\beta}^{(r-1)}\| \to 0$ **do**
  1) <u>Output Messages Step:</u>
  for $t = 1$ **to** T **do**
    $\widehat{c}_t^{(r)} = \sum_{j=1}^{q} \mathcal{X}_{t,j} \widehat{\beta}_j^{(r-1)} - \widehat{s}_t^{(r-1)} * \widehat{\tau}_t^{c,(r)}$
    $\widehat{\tau}_t^{c,(r)} = \sum_{j=1}^{q} \mathcal{X}_{t,j}^2 \widehat{\tau}_j^{\beta,(r-1)}$
    $\widehat{s}_t^{(r)} = g_{out}\left(\widehat{c}_t^{(r)}, \widehat{\tau}_t^{c,(r)}, y_t\right)$
    $\widehat{\tau}_t^{s,(r)} = -\frac{\partial}{\partial \widehat{c}} g_{out}\left(\widehat{c}_t^{(r)}, \widehat{\tau}_t^{c,(r)}, y_t\right)$
  end for
  2) <u>Input Messages Step:</u>
  for $j = 1$ **to** q **do**
    $\widehat{d}_j^{(r)} = \widehat{\beta}_j^{(r-1)} + \widehat{\tau}_j^{d,(r)} \sum_{t=1}^{T} \mathcal{X}_{t,j} \widehat{s}_t^{(r)}$
    $\widehat{\tau}_j^{d,(r)} = \left(\sum_{t=1}^{T} \mathcal{X}_{t,j}^2 \widehat{\tau}_t^{s,(r)}\right)^{-1}$
    $\widehat{\beta}_j^{(r)} = g_{in}\left(\widehat{d}_j^{(r)}, \widehat{\tau}_j^{d,(r)}\right)$
    $\widehat{\tau}_j^{\beta,(r)} = \widehat{\tau}_j^{d,(r)} \frac{\partial}{\partial \widehat{d}} g_{in}\left(\widehat{d}_j^{(r)}, \widehat{\tau}_j^{d,(r)}\right)$
  end for
  $r = r + 1$
end while
Obtain mean and variance of β as $\widehat{\beta} = \left(\widehat{\beta}_1^{(r)}, \ldots, \widehat{\beta}_q^{(r)}\right)$ and $\tau^{\beta} = \left(\widehat{\tau}_1^{\beta,(r)}, \ldots, \widehat{\tau}_q^{\beta,(r)}\right)$

Why is GAMP efficient?

- The algorithm iterates through computationally trivial scalar multiplications and additions
- Worst case complexity of only $\mathcal{O}(Tq)$
- At its core are the scalar functions g_{in} and g_{out}
- Exact form of these two functions depends on the form of the prior distribution and the likelihood
- Regardless of the form of the nonlinear scalar functions g_{in} and g_{out}, the worst-case complexity of the GAMP algorithm is always $\mathcal{O}(Tq)$
- Final product are $\widehat{\mu}_{\beta,j}$ and $\widehat{\tau}_{\beta,j}$, posterior mean and variance of β_j, respectively
- Prior hyperparameters can be updated using EM updates within each GAMP iteration
- Conditional on $\widehat{\mu}_{\beta,j}$ and $\widehat{\tau}_{\beta,j}$, stochastic volatility is updated using the simple estimator presented previously

Summary of what I have presented so far

1. Interested in time-varying parameter regression
2. View time-varying parameter estimation problem as a high-dimensional regression and tackle it using Bayesian shrinkage
3. View stochastic volatility estimation problem as equivalent problem of estimating an unknown (static) mean
4. Use a highly efficient message passing algorithm for inference and develop it to work with the specific shrinkage prior and stochastic volatility estimator

- Next I am presenting empirical results, but I won't surprise you (typical macro forecasting exercise using various TVP models)
- GAMP-TVP model has $Tq \approx 700 \times 40 \approx 30,000$ predictors
- In practice could have gone with non-macro application that has millions of predictors (but I like macro!)

Forecasting US inflation

- The methodology is used to forecast inflation with time-varying parameters and predictors
- It is now established that for forecasting US inflation, both stochastic volatility and regression coefficients are needed
- Very popular: Stock and Watson (2007, JMCB), TV intercept, no predictors
- Large recent literature on algorithms for efficient estimation of high-dimensional TVP models for inflation
- Kalli and Griffin (2014, JoE), Koop and Korobilis (2012, IER), Groen et al (2012, JBES), Chan et al (2013, JBES), Nakajima and West (2013, JBES), Giordani and Kohn (2008, JBES) among others

→ All methodologies use quarterly data, and few predictors

Data

- No surprise on the data front: FRED-MD dataset
- Forecasting total CPI, PCE deflator (results same for other measures)
- Monthly data from 1959M1 to 2016M6, total of 115 series
- Forecasting model follows Stock and Watson (1999, JME):

$$\pi_{t+h} - \pi_t = \phi_{t,0} + f_t\theta_t(L) + \Delta\pi_t\gamma_t(L) + e_{t+h}, \qquad (33)$$

  where $e_t \sim N(0, \sigma_t^2)$ and f_t are factors
- Specification implies inflation is I(1)
- Forecasts for $h = 1, 3, 6, 12$ months ahead
- Forecasts evaluated at last 50% of the sample, MSFEs and APLs relative (ratio and spread, respectively) to AR(2) model.

Competing models

- **KP-AR:** Koop and Potter (2007) structural breaks AR(p) model;
- **GK-AR:** Giordani and Kohn (2008) structural breaks AR(p) model;
- **TVP-AR:** Pettenuzzo and Timmermann (2017) time-varying parameter AR(p) model;
- **UCSV:** Stock and Watson (2007) unobserved components stochastic volatility;
- **TVD:** Chan et al. (2012) time-varying dimension regression
- **TVS:** Kalli and Griffin (2014) time-varying sparsity regression
- **BMA:** George and McCulloch (1993) stochastic search variable selection regresison
- **TVP-BMA:** Groen et al. (2012) time-varying Bayesian model averaging model

Point forecast evaluation

Table: Forecast performance (MSFEs)

	CPI			
	$h = 1$	$h = 3$	$h = 6$	$h = 12$
KP-AR	0.970	0.879	0.849***	0.834***
GK-AR	0.999	1.008	1.009	1.005
TVP-AR	0.949	0.867***	0.828***	0.837***
UCSV	1.027	0.970	0.911**	0.916*
TVD	0.957	0.867***	0.862***	0.850***
TVS	1.175	0.960	0.963	1.005
BMA	0.982*	0.588***	0.542***	0.531***
TVP-BMA	1.090	0.770***	0.772**	0.629**
TVP-GAMP	0.923**	0.461***	0.421***	0.413***
	PCE deflator			
	$h = 1$	$h = 3$	$h = 6$	$h = 12$
KP-AR	1.018	0.845***	0.806***	0.783***
GK-AR	0.999	0.996	1.005	0.999
TVP-AR	1.010	0.793***	0.720***	0.732***
UCSV	1.064	0.841***	0.810***	0.761***
TVD	1.015	0.787***	0.744***	0.742***
TVS	1.041	0.857***	0.817***	0.814***
BMA	1.014	0.713***	0.663***	0.654***
TVP-BMA	1.158	0.842**	0.798*	0.812
TVP-GAMP	0.982	0.614***	0.584***	0.565***

Density forecast evaluation

Table: Forecast performance (APLs)

	CPI			
	$h = 1$	$h = 3$	$h = 6$	$h = 12$
KP-AR	0.060	0.135	-0.006	0.023
GK-AR	-0.027	0.033	0.025	-0.027
TVP-AR	0.216	0.095	0.045	0.071
UCSV	0.184	0.031	0.033	-0.002
TVD	-8.107	-2.665	-1.862	-1.859
TVS	0.032	0.154	0.100	0.058
BMA	0.019	0.303	0.279	0.292
TVP-BMA	0.149	0.394	0.379	0.358
GAMP	0.017	0.528	0.422	0.381
	PCE deflator			
	$h = 1$	$h = 3$	$h = 6$	$h = 12$
KP-AR	-0.033	0.071	0.044	0.016
GK-AR	-0.066	0.000	0.009	0.009
TVP-AR	0.068	0.157	0.116	0.118
UCSV	0.051	0.065	0.062	0.081
TVD	-9.103	-2.887	-1.784	-1.559
TVS	0.004	0.149	0.167	0.103
BMA	-0.035	0.203	0.211	0.203
TVP-BMA	0.024	0.277	0.323	0.290
GAMP	0.046	0.258	0.279	0.266