Essex Finance Centre

Working Paper Series

Working Paper No18: 04-2017

"Forecasting with many predictors using message passing algorithms"

"Dimitris Korobilis"

Essex Business School, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ
Web site: http://www.essex.ac.uk/ebs/

Forecasting with many predictors using message passing algorithms

Dimitris Korobilis
University of Essex

May 6, 2017

Abstract

Machine learning methods are becoming increasingly popular in economics, due to the increased availability of large datasets. In this paper I evaluate a recently proposed algorithm called *Generalized Approximate Message Passing (GAMP)*, which has been popular in signal processing and compressive sensing. I show how this algorithm can be combined with Bayesian hierarchical shrinkage priors typically used in economic forecasting, resulting in computationally efficient schemes for estimating high-dimensional regression models. Using Monte Carlo simulations I establish that in certain scenarios GAMP can achieve estimation accuracy comparable to traditional Markov chain Monte Carlo methods, at a tiny fraction of the computing time. In a forecasting exercise involving a large set of orthogonal macroeconomic predictors, I show that Bayesian shrinkage estimators based on GAMP perform very well compared to a large set of alternatives.

Keywords: high-dimensional inference; compressive sensing; belief propagation; Bayesian shrinkage; dynamic factor models

JEL Classification: C11, C13, C15, C22, C52, C53, C61

1 Introduction

Increased availability of large datasets is both a blessing and a curse for macroeconomists and financial analysts. It is a blessing because we currently have so many advanced statistical methods and computational tools that allow us to uncover new stylized facts from disaggregated and other high-dimensional datasets, that economists of the past didn't have the chance to consider. It is a curse because policy-makers and analysts cannot any more make decisions based on a handful of preferred indicators, rather, they have to monitor hundreds or thousands of variables, which can be costly to maintain and hard to interpret. As a consequence, given that data availability increases at a polynomial rate, a major challenge for modern economists and econometricians is to design estimation algorithms that i) are able to separate the "signal" from the "noise" in a high-dimensional dataset, and ii) are computationally efficient in order to support real-time decision making by policy-makers and analysts. The purpose of this paper is to introduce to the economics literature a machine learning algorithm that can be used to perform dynamic programming, maximum likelihood estimation, and Bayesian inference using shrinkage priors. The so-called Generalized Approximate Message Passing (GAMP) has been proposed by Rangan (2011), and is an algorithm that extends the Approximate Message Passing (AMP) algorithm of Donoho, Maleki and Montanari (2009). In turn these are fast, approximate versions of general message passing algorithms used in machine learning; see Barber (2012) for an insightful introduction to this literature.

There is more than one way to motivate message passing methods. For example, Donoho, Maleki and Montanari (2011) motivate AMP as an iterative thresholding algorithm that is a fast alternative to convex optimization and linear programming methods; see also Bayati and Montanari (2011). In fact, all variants of message passing algorithms are dynamic programming methods designed for efficiently performing large computations by distributing calculations among a number of simpler processors. Here, I build on Rangan (2011) and present an intuitive derivation of generalized approximate message passing based on graphical methods. My starting point is a Bayesian factor graph (Kschischang, Frey and Loeliger, 2001), that is, a

graphical representation of the posterior probability function of the regression coefficients using nodes and edges. Such graphical representation of a probabilistic model allows factorizations (decompositions) of the parameter posterior distribution into simpler probability functions. These factorizations are basically probability rules that define how a "message is passed" from one node of the graph to the next. The basic idea behind such decompositions in factor graphs is that of the distributive law:

$$a \cdot b + a \cdot c = a(b + c). \tag{1}$$

In terms of the variables a, b, c, the axiom above implies one less operation on the right hand side. In terms of probability distributions we can derive similar factorizations that could result - in high-dimensional spaces - in huge savings in computing power.

It becomes, therefore, apparent that GAMP can be thought of as an algorithm that breaks down the problem of estimating a high-dimensional parameter vector into smaller, more accessible problems. This is achieved by factorizing the high-dimensional Bayesian parameter posterior into approximate, analytical expressions for the marginal parameter posteriors. Put differently, if one is faced with the problem of estimating a p-dimensional vector of parameters β, GAMP allows to obtain approximate expressions for the posteriors of each element β_i for $i = 1, ..., p$[1]. While the next section outlines the exact mechanics behind such approximations, three questions immediately arise: i) how accurate these approximations are, ii) how adaptable and versatile the algorithm is in various modelling scenarios, and iii) what are the computational gains compared to existing algorithms? This paper aims to give detailed answers to all these questions.

First I show that GAMP can be combined with, among others, Normal-Gamma shrinkage priors (Griffin and Brown, 2010) as well as variable selection priors (George and McCulloch, 1993). While GAMP only provides the solution to sparse regression coefficients, any additional parameters involved in an econometric model, such as a regression variance, can be updated using the EM algorithm. An extension of GAMP in heteroskedastic regression models is also

[1]The exact marginal posterior of β_i involves calculating a $p-1$ multidimensional integral, since this posterior is obtained after "marginalizing" the impact of the remaining $p-1$ elements of β. When p is large, computing such integrals is impossible, even for simple regression models.

presented. Next, I perform a Monte Carlo experiment where I generate data from sparse regression models in order to evaluate the numerical stability and computational efficiency of GAMP. GAMP-based algorithms perform at least as well, or better in several cases, than MCMC-based shrinkage algorithms[2] in recovering the true regression coefficients, at a fraction of the computing time. Finally, the precision and usefulness of GAMP is evaluated in an extensive forecasting exercise following closely Stock and Watson (2012). This exercise involves estimating univariate regression models for forecasting 222 quarterly US macroeconomic series using a large set of orthogonal predictors, typically in the form of factors (principal components). Under a wide-range of alternative shrinkage algorithms (Bayesian MCMC-based algorithms, bootstrap aggregation, and naive approaches) the proposed GAMP-based algorithms perform extremely well. The results are robust both when looking at all 222 series as well as a selection of 14 important series (such as GDP, employment, IP, CPI inflation, Federal funds rate, SP500 returns etc), and under both recursive and rolling forecasting schemes. Therefore, this paper makes a strong case in favor of establishing GAMP-based algorithms as a significant and "low-cost" (in terms of requirements in tuning, computational resources, and maintenance/updating) tool for macroeconomic forecasting using many predictors, and for general monitoring of large datasets.

In the next section I outline this new methodology using Bayesian arguments, and I explain the intuition behind the approximations to marginal posterior distributions achieved using GAMP. I also discuss convergence issues and demonstrate the kind of shrinkage and variable selection priors one can combine with GAMP. Next, in section 3 I provide results of the Monte Carlo study, and in section 4 I implement the large-scale forecasting exercise. Section 5 concludes the paper with an evaluation of GAMP and with thoughts for future incorporation of GAMP in other models and settings that economists are interested in (such as vector autoregressions).

[2]In particular, I contrast to the Bayesian lasso (Park and Casella, 2008) and stochastic search variable selection (SSVS; George and McCulloch, 1993).

2 Methodology

The starting point is the following regression model in vector/matrix form

$$y = x\beta + \varepsilon \tag{2}$$

where y is a $T \times 1$ vector of the variable of interest, x is a $T \times p$ matrix of predictors (possibly including lags of y), β is a $p \times 1$ vector of coefficients, and $\varepsilon \sim N\left(0, \sigma^2 I_T\right)$. A motivating assumption for considering estimation algorithms that perform shrinkage and are computationally efficient is that $p \gg T$, that is, the specification in equation (2) can be thought of as a regression with more predictors than observations. However, I will subsequently show that GAMP performs very well in any "large p" regression case, even if $p < T$. Note that when estimating a regression interest also lies in the estimation of σ^2, but I first consider this parameter to be known and I subsequently discuss joint estimation of β, σ^2 later in this section.

Let us consider first an independent (but not necessarily i.i.d) prior for β, denoted $p(\beta) = \prod_{i=1}^{p} p(\beta_i)$, and the resulting posterior from Bayes Theorem

$$\begin{aligned} p(\beta|y) &\propto p(y|\beta)\, p(\beta) && (3) \\ &= \prod_{t=1}^{T} p(y_t|\beta) \prod_{i=1}^{p} p(\beta_i). && (4) \end{aligned}$$

The exact marginal posterior for β_i, $i = 1, ..., p$ is of the form

$$\begin{aligned} p(\beta_i|y) &= \int p(\beta|y)\, d\beta_{j\neq i}, && (5) \\ &\propto \int p(y|\beta)\, p(\beta)\, d\beta_{j\neq i}, && (6) \\ &= p(\beta_i) \int p(y|\beta) \prod_{j=1, j\neq i}^{p} p(\beta_j)\, d\beta_{j\neq i}, && (7) \end{aligned}$$

where $d\beta_{j\neq i}$ denotes integration over the whole set of $p-1$ parameters β_j for $j \neq i$. Therefore, the formula above requires integration over a $(p-1)$-dimensional integral, a numerical problem

that can become computationally infeasible for high-dimensional vectors β.

2.1 Generalized Approximate Message Passing

The computationally efficient GAMP algorithm approximations can be motivated using factor graphs, that is, graphical models that allow to factorize random variables into lower-dimensional quantities. In our case, the random variable we want to factorize is the high-dimensional parameter posterior β. Based on Bayes Theorem above, the resulting factor graph is depicted in Figure 1. This graph consists of variable vertices $\beta = (\beta_1, ..., \beta_p)$ which are denoted with a white circle, and function vertices $f = [p(\beta), p(y|\beta)] = [p(\beta_1), ..., p(\beta_p), p(y_1|\beta), ..., p(y_T|\beta)]$ which are represented using filled boxes. I denote by $\mu_{p(\bullet)\to a}$ the message passed from probability function $p(\bullet)$ to random variable a, and vice-versa for the message $\mu_{a\to p(\bullet)}$. We are now able to redefine the marginal posterior of β_i, presented in equation (7), as the product of incoming messages at node β_i in the graph

$$p(\beta_i|y) = \mu_{p(\beta_i)\to\beta_i} \prod_{t=1}^{T} \mu_{p(y_t|\beta)\to\beta_i}. \tag{8}$$

The message $\mu_{p(\beta_i)\to\beta_i}$ is simply the prior distribution $p(\beta_i)$. According to the *sum-product message rule*, the term inside the product can be decomposed as

$$\mu_{p(y_t|\beta)\to\beta_i} = \int p(y_t|\beta) \prod_{j=1, j\neq i}^{p} \mu_{\beta_j\to p(y_t|\beta)} d\beta_{j\neq i}. \tag{9}$$

In the decomposition above, the message from node β_j to function $p(y_t|\beta)$ is the product of all incoming messages to node β_i, excluding the message coming from $p(y_t|\beta)$ itself

$$\mu_{\beta_j\to p(y_t|\beta)} = p(\beta_j) \prod_{s=1, s\neq t}^{T} \mu_{p(y_s|\beta)\to\beta_j}. \tag{10}$$

We can see in equations (9)-(10) that in order to obtain the message $\mu_{p(y_t|\beta)\to\beta_i}$ we need $\mu_{\beta_j\to p(y_t|\beta)}$ and vice-versa. Therefore, one can simply update both equations iteratively, using a scheme that is called Belief Propagation (BP; see Pearl, 1982) and consists of the following

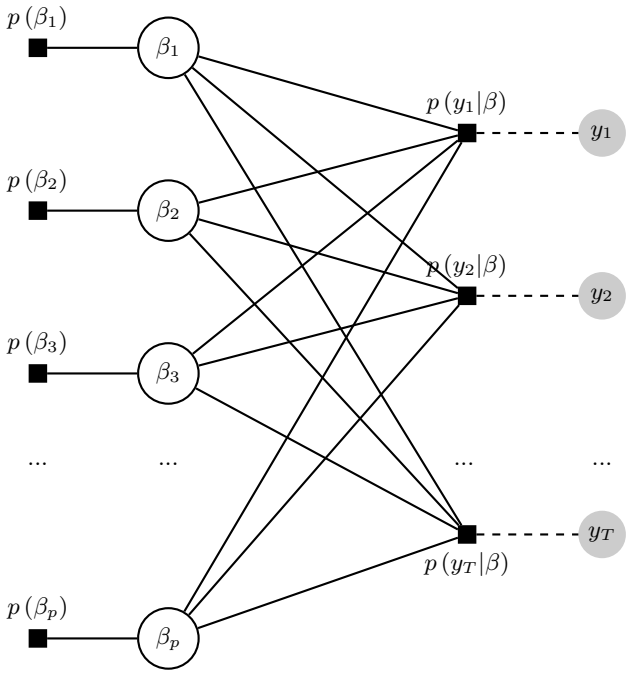

Figure 1: Factor graph for the posterior distribution of β

iterations

$$\mu^{(r+1)}_{p(y_t|\beta)\to\beta_i} = \int p(y_t|\beta) \prod_{j=1,j\neq i}^{p} \mu^{(r)}_{\beta_j\to p(y_t|\beta)} d\beta_{j\neq i}, \tag{11}$$

$$\mu^{(r+1)}_{\beta_j\to p(y_t|\beta)} = p(\beta_j) \prod_{s=1,s\neq t}^{T} \mu^{(r)}_{p(y_s|\beta)\to\beta_j}, \tag{12}$$

where superscript (r) denotes the r^{th} iteration of the algorithm. In graphs with a tree structure, one iteration of the algorithm above can recover the exact marginal posteriors for the parameters β_i. In a factor graph with loops, that is, when a message can start from a given node and return to the same node using various alternative paths, the BP algorithm can sometimes achieve a good approximation. This algorithmic issue is discussed in the following subsection.

From that point onward the GAMP algorithm can be derived by introducing two further approximations to the BP iterations. First, when $p \to \infty$ a central limit theorem

(CLT) postulates that the messages $\prod_{j=1,j\neq i}^{p} \mu_{\beta_j \to p(y_t|\beta)}$ can be approximated by a Gaussian distribution with respect to the uniform norm[3]. This result means that both messages in (9)-(10) can be represented as products of Gaussian distributions. A second approximation involves taking the Taylor-series expansion of terms in the messages, so that the mean and variance of $p(\beta_i|y)$ can be obtained analytically up to the omission of $O(1/p)$ terms. Exact derivation of these steps involve many tedious steps, and the reader is referred to the Technical Appendix of Rangan (2011) for more details. What is important to stress at this point is that both the CLT and Taylor-series approximations vanish as $p \to \infty$ with $p/T \to \delta$ for some constant δ; see Rangan (2011) for more details. This is an example of the "blessing of Big Data", and the GAMP algorithm fully facilitates the large p asymptotics.

The final product of all the approximations to the two Belief Propagation update rules of equations (11) - (12), is a simple iterative algorithm that provides marginal parameter posteriors of β_i and z_t, where define $z_t = x_t\beta$ and assume, without loss of generality[4], that $x_t \sim N\left(0, T^{-1}I\right)$. In particular, GAMP approximates $p(\beta_i|y)$ with

$$p(\beta_i|y, q_i, \tau_{q,i}) = \frac{p(\beta_i)\, N(\beta_i|q_i, \tau_{q,i})}{\int_\beta p(\beta_i)\, N(\beta_i|q_i, \tau_{q,i})}, \tag{13}$$

where $q_i, \tau_{q,i}$ are certain quantities computed iteratively, and are given in the Technical Appendix. Similarly, the second update rule of Belief Propagation provides an approximation to $p(z_t|y)$ using

$$p(z_t|y, c_t, \tau_{c,t}) = \frac{p(y_t|z_t)\, N(z_t|c_t, \tau_{c,t})}{\int_z p(y_t|z_t)\, N(z_t|c_t, \tau_{c,t})}, \tag{14}$$

where $c_t, \tau_{c,t}$ are also computed iteratively and more details can be found in the Technical Appendix.

[3]This is a result of the Berry-Esseen central limit theorem which states that a sum of random variables converge to a Gaussian density; see a proof of that theorem in Donoho, Maleki and Montanari (2010). Given that the Belief Propagation equations involve products of random variables, rather than sums, derivations of GAMP based on this central limit theorem typically proceed by taking logarithms of equations (8)-(10). The marginal posterior $p(\beta_i|y)$ is then recovered by performing an exponential transformation of the log messages, and by normalizing so that the posterior integrates to one.

[4]As Donoho, Maleki and Montanari (2010) show, one can relax this assumption and consider other distributions; see also Bayati and Montanari (2011).

2.2 Stability of GAMP

The GAMP algorithm, whose steps are provided in detail in the Technical Appendix, iterates through simple scalar multiplications and additions that result in a total of $\mathcal{O}(Tp)$ algorithmic operations. That is, estimation of the marginal parameter posterior distribution does not involve any matrix inversion that is typically met in many forms of parametric estimation algorithms. Convergence is achieved when the difference between estimates of the posterior mean of β between two consecutive iterations is below a pre-specified tolerance level. Therefore, an important question is whether there are theoretical guarantees that the algorithm will converge. The answer is partly yes. In graphs with loops, such as the one specified in the previous section, the Belief Propagation algorithm will approximate the marginal posterior distribution for β_i by only considering its dependence on its direct neighbours. What this means in practical terms is that all the factorizations of distributions resulting from BP implicitly assume that there is a low correlation between the p elements β and, as a consequence, the p columns of X. In general, many MCMC-based variable selection and shrinkage algorithms also suffer from the presence of highly correlated predictors. However, GAMP, due to its underlying approximate factorizations, has even lower threshold for handling correlated data. In the next section I show using Monte Carlo experiments that GAMP in the case of regressions with mildly correlated predictors can achieve estimation accuracy comparable to MCMC, at a fraction of the computing time [5]. Additionally, the empirical exercise involves regressions with orthogonal predictors such as principal components, or predictors that have been orthogonalized (that is, rotated and rescaled such that $x \sim iid(0, I_p)$) prior to estimation.

2.3 Joint Parameter Estimation

A salient feature of the derivations above is that they are also valid when the prior on β is conditional on some hyperparameters θ, that is when the prior is of the general hierarchical

[5] Additionally, there is a large literature in engineering documenting that loopy BP works very well in various applications including turbo decoding (McEliece et al., 1998), computer vision (Freeman et al., 2000), and compressed sensing (Baron et al., 2010).

form

$$p(\beta, \theta) = p(\beta|\theta)\, p(\theta). \tag{15}$$

In this paper I consider two cases of hierarchical priors that result in regularized posterior means. The first prior considered is a Normal-Gamma prior (Griffin and Brown, 2010) of the form

$$\begin{aligned} p(\beta_i|\alpha_i) &= N\left(0, \alpha^{-1}\right), & (16)\\ p(\alpha_i) &= Gamma(\underline{a}, \underline{b}). & (17) \end{aligned}$$

I use here the convention that prior hyperparametes with an underscore are fixed (calibrated) while prior hyperparameters without an underscore are random variables and are, hence, updated by the data. With this hierarchical prior specification each prior variance α_i^{-1} is learned by the data, while following arguments in Griffin and Brown (2010) it can be shown that this prior leads to sparse solutions by allowing $alpha_i \to \infty$. Following Zou, Li, Fang and Li (2016) the estimation algorithm under this prior is hereafter denoted as Sparse Bayesian Learning (SBL).

The second prior considered is the "spike and slab" prior, which is typically used in Bayesian inference for the purpose of variable selection and for testing the hypothesis $H_0 : \beta_i = 0$, and which is of the form

$$\begin{aligned} p(\beta_i|\pi_0) &= (1-\pi_0)\,\delta_0 + \pi_0 N\left(0, \alpha^{-1}\right), & (18)\\ p(\pi_0) &= Beta\left(\underline{\rho}_1, \underline{\rho}_2\right). & (19) \end{aligned}$$

This is a mixture prior with one component being a point mass at zero (the Dirac delta function δ_0, which can be thought of as the limit of a $N(0,0)$ distribution), and the other component being a typical Normal prior with $\alpha_i^{-1} \neq 0$. From this point forward the abbreviation SNS (Spike N' Slab) will be used to denote the resulting GAMP algorithm using this prior. Finally, I also consider a prior for the regression variance, which in the previous subsections I intentionally

ignored and considered given. The prior considered for this parameter is also of standard form, that is

$$p\left(\sigma^2\right) = Gamma\left(\underline{c}_1, \underline{c}_2\right). \tag{20}$$

Updates of prior hyperparameters as well as the regression variance can be implemented by combining the GAMP algorithm with Expectation-Maximization (EM) updates. In the Technical Appendix I illustrate that derivations of these steps result in familiar formulas from MCMC updates. Additionally, there are several papers in the literature establishing that such joint EM-GAMP algorithms result in consistent estimates of β; see further results in Kamilov, Rangan, Fletcher and Unser (2014).

3 Simulation study

In this section I compare the performance of GAMP-based algorithms using data generated from sparse regression models, and contrast them to MCMC-based algorithms. I generate p predictors, with T observations each, from a Normal distribution with correlation $corr(x_i, x_j) = \rho^{|i-j|}$ for $\rho \in [0, 1]$. I assume that only q columns of the predictors x are important for y, and the remaining $p-q$ columns are excluded from the regression. That is, the coefficient vector is of the form $\beta = [\beta_1, ..., \beta_q, 0, 0, ..., 0, 0]$, where $q = \lfloor c \times p \rceil$, $0 < c < 1$ and $\lfloor \bullet \rceil$ denotes round-off to the nearest integer. I generate β_i for $i = 1, ..., q$ from a continuous $U(-4, 4)$ distribution.

I consider various combinations of p, T and ρ cases, in order to assess how sparsity, correlation, and changing number of samples impact performance:

1. Model 1: $T = 50$, $p = 100, 200, 500$ and $\rho = 0.3$. I assume moderate correlation, and the case of a small sample T that allows us to fully understand how the algorithm works in the large p - small T limit. I set $c = 0.01$ meaning only one, two and five predictors out of $p = 100, 200, 500$, respectively, are responsible for having generated y.

2. Model 2: $T = 200$, $p = 100, 200, 500$ and $\rho = 0.3$. This is like Model 1, but with higher number of observations, to reflect approximate the T found in quarterly macroeconomic

data. I set $c = 0.05$ meaning only one predictor out of p is responsible for having generated y. I set $c = 0.05$ meaning only five, 10 and 25 predictors out of $p = 100, 200, 500$, respectively, are responsible for having generated y.

3. Model 3: $T = 200$, $p = 100$ and $\rho = 0.9$. In this case I only evaluate the case where $T > p$ so that orthogonalization of predictors is possible. For such high correlation, the GAMP algorithm would collapse and the MCMC-based estimation algorithms would also have a hard time converging. Orthogonalization is implemented by simply taking the sample covariance of the predictors x, $\widehat{\Omega}$ and defining $\widetilde{x} = x\widehat{W}^{-1}$ where W is the Cholesky factor of $\widehat{\Omega}$. Note that if $p > T$ then W is rank deficient and orthogonalization of the original space spanned by the columns of x is not possible. I also set $c = 0.05$ in this case.

Each of the three Monte Carlo simulations is repeated 500 times. Precision of each algorithm is measured by means of absolute deviations of estimated coefficients from the actually generated ones in all 500 cases. That is, the estimate with the smallest value of the mean absolute deviation statistic

$$MAD = \frac{1}{500}\sum_{i=1}^{500}\left|\widehat{\beta}_{M_j}^{(i)} - \widetilde{\beta}^{(i)}\right|,$$

where $\widehat{\beta}_{M_j}^{(i)}$ is the point estimate (or posterior mean) of β from estimation method M_j, and $\widetilde{\beta}^{(i)}$ are the generated coefficients in each Monte Carlo iteration. I evaluate the following four algorithms: 1) the GAMP algorithm with Normal Gamma prior, which I denote as sparse Bayesian learning (SBL); 2) the GAMP algorithm with spike and slab (SNS) prior; 3) the Bayesian least absolute shrinkage and selection operator (lasso) estimated using the Gibbs sampler as in Park and Casella (2008); and 4) the stochastic search variable selection (SSVS) algorithm of George and McCullogh (1993), also based on the Gibbs sampler. Therefore, in this comparison there are four models M_j for $j = SBL, SNS, LASSO, SSVS$.

The following Figures 2-4 show boxplots of the MAD statistics for the four algorithms run over the 500 Monte Carlo iterations for the three exercises, respectively. Note first that boxplots of the MAD statistics of a naive, unrestricted estimator (OLS applied using one predictor at a time) is omitted from the three graphs, because these are typically five to 20 times larger

(depending on the values of T, p) than the MAD values of the four shrinkage algorithms[6]. So the first general observation is that, even if there are visual differences among the distribution of MAD statistics for the four estimators, these are typically small if the MAD of OLS applied predictor-by-predictor is used as a reference point. When T is small relative to p, as depicted in Figure 2 for the case $T = 50$, the SBL and SNS versions of the GAMP algorithm perform well. On average they give sharper results compared to LASSO and SSVS, with lower averages and more concentrated distributions of MAD statistics. When T is larger in which case a different ratio δ is achieved (see discussion in subsection 2.1) then performance of the four algorithms is comparable when $p \leq T$, i.e. for $p = 100, 200$. For the case $p = 500$ the two GAMP-based algorithms perform slightly worse than the two MCMC-based algorithms, but differences can be considered small (the average MAD of OLS in this case is 0.67, while for the four shrinkage algorithms all average MADs are less than 0.05). In general, under mild correlation among predictors, GAMP-based algorithms perform comparably to MCMC-based algorithms.

An interesting case is the one where one wants to do inference with correlated predictors. Figure 4 shows what happens in the case $T = 200, p = 100$. While the assumption $p < T$ seems quite restrictive for general Big Data applications, the example in Model 3 is a quite realistic representation of many modern macroeconomic applications [7]. In this instance also the performance of GAMP methods is excellent and considerably better than that of the LASSO.

[6] In particular, for the mildly correlated predictors that are generated, the OLS applied predictor-by-predictor performs as well (in terms of MADs) for the q coefficients which are non-zero. It is for the case of the zero coefficients where the shrinkage estimators generate substantially lower error compared to OLS.

[7] For example, Stock and Watson (2012) and Korobilis (2013) consider datasets with around $p = 100$ predictors and $T = 200$ quarterly observations.

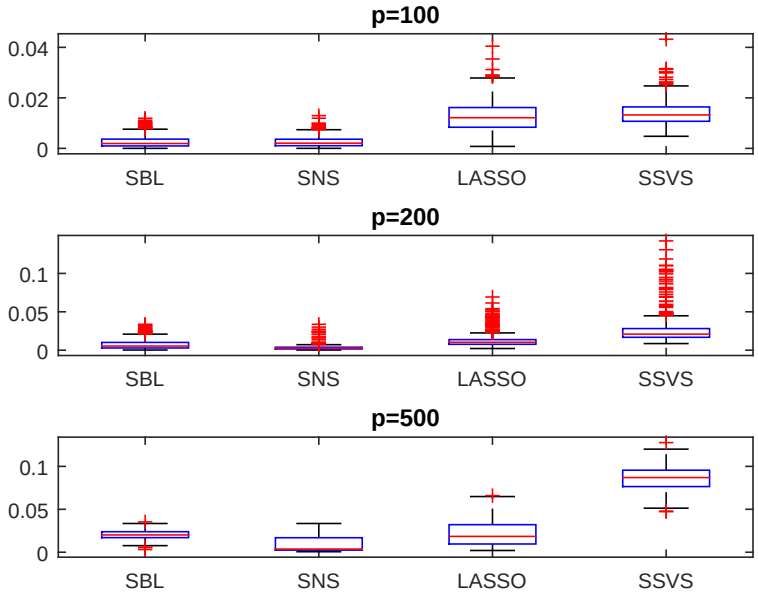

Figure 2: Boxplots of MAD statistics over the 500 Monte Carlo iterations for Model 1 case ($T = 50$, $p = 100, 200, 500$).

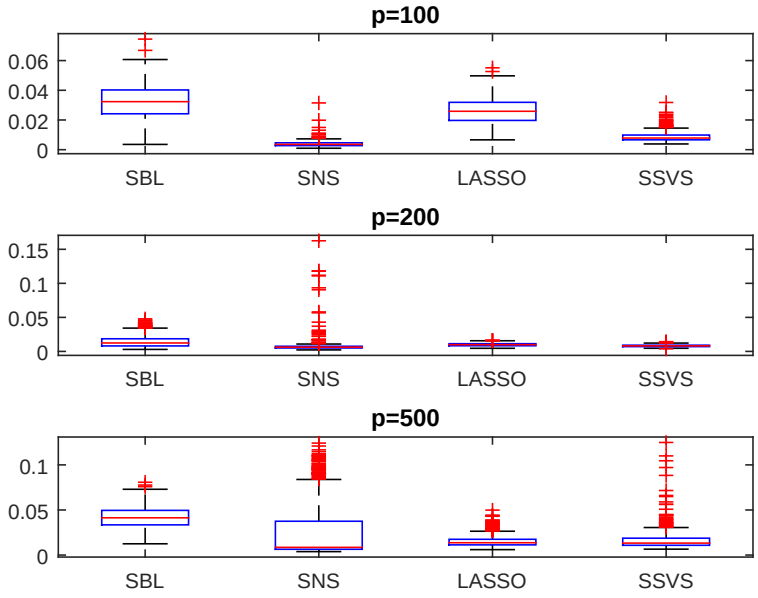

Figure 3: Boxplots of MAD statistics over the 500 Monte Carlo iterations for Model 2 case ($T = 200$, $p = 100, 200, 500$).

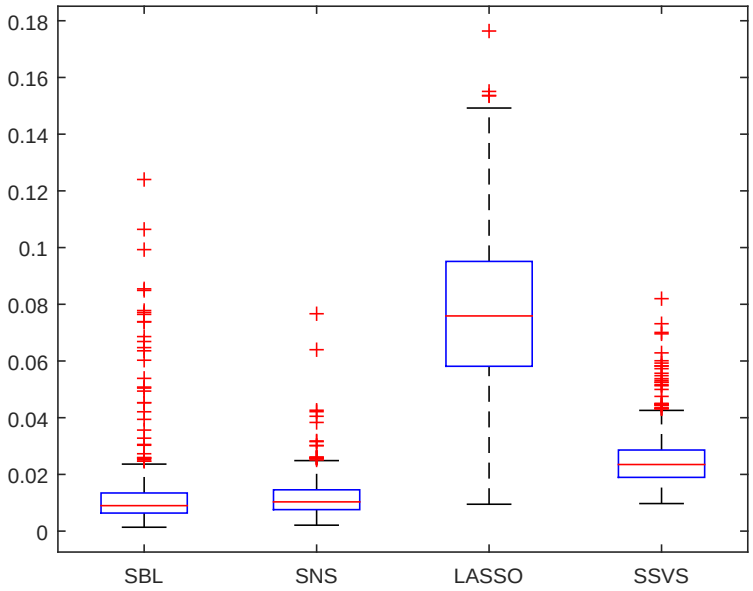

Figure 4: Boxplots of MAD statistics over the 500 Monte Carlo iterations for Model 3 case ($T = 200$, $p = 100$).

Having established that GAMP-based methods perform comparably when the data generating process is that of a sparse regression model, I proceed to documenting the vast computational gains from the GAMP approximations. Table 1 demonstrates a striking feature of GAMP, that is, the fact that it is much faster than both MCMC-based algorithms used in the simulations above. Entries in this Table are computing times measured in seconds (using the `tic` and `toc` commands in MATLAB). The differences are vast, since for the majority of the experiments the GAMP-based algorithms remain below 0.01 seconds. The only exception is for SNS when $T = 200$ and $p = 500$, and the explanation is that these algorithms run until a convergence criterion is achieved (as is the case for e.g. the EM algorithm) and in this case convergence took longer. In contrast, MCMC based algorithms have to run for a fixed number of iterations [8]. Finally, note that while the GAMP algorithm can be further

[8]Note that for the purposes of the Monte Carlo exercise in particular, I run both MCMC-based estimators using only 2000 iterations after discarding an initial 1000 burn-in iterations. This low number ensures satisfactory numerical precision, but in practical situations one would want to run thousand times more MCMC iterations. Thus, the high computing times of LASSO and SSVS are actually favourable due to the limited number of iterations.

enhanced by trivially modifying it to run in multiple CPU cores, MCMC-based algorithms are not parallelizable unless further approximations are introduced.

Table 1: 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.

4 Empirics

4.1 Forecasting model, data, and competing methods

For the purposes of forecasting, the regression in equation (2) is converted into a forecasting relationship for y by determining appropriately the timing of the predictors. Therefore, the following regression is defined

$$y_{t+h} = y_t\phi + P_t\beta + \varepsilon_{t+h}, \tag{21}$$

where y_{t+h} is the h-step ahead value of the variable of interest and P_t are orthogonal predictors, which, following Stock and Watson (2012), are going to be factors from a large macroeconomic dataset. The dataset consists of 222 quarterly U.S. macroeconomic and financial time series observed for the period 1959Q1 - 2015Q3. The series are transformed by taking logarithms and/or differencing, that is, first differences of logarithms are used for real quantity variables, first differences are used for nominal interest rates, and second differences of logarithms for price series. When a given variable is the variable to be predicted, y^h_{t+h}, then relevant h-step ahead transformations are defined, see also Stock and Watson (2012) for more details.

All 222 macroeconomic variables are used, one at a time, as the variable of interest, that is, the variable to forecast. The remaining variables are used as exogenous right-hand side

predictors. In particular, when a high-level aggregate and a subaggregate variable are present, and when these two are typically related by an identity, I only use the lower-level subaggregate to extract factors. This leaves 130 variables to extract the factors from, despite the fact that we evaluate forecasts of all 222 variables. Out of a maximum of 130 factors, I follow Stock and Watson (2012) and use a maximum of 50 factors estimated by simple principal component analysis.

The reqression in equation (21) is estimated using several competing estimators. I contrast the two GAMP-based shrinkage estimators, **SBL** and **SNS** described in subsection 2.3, with the following cases

1. Bayesian lasso prior estimated with MCMC (**LASSO**): More description of this prior and its settings is provided in the previous section and the Technical Appendix.

2. Stochastic search variable selection (**SSVS**): More description of this prior and its settings is provided in the previous section and the Technical Appendix.

3. Bayesian Model Averaging (**BMA**): This setting follows Fernandez, Ley and Steel (2001a,b). A g-prior is specified with $g = 1/p^2$, where p is the number of predictors. Estimation is via MCMC methods and in particular the MC^3 algorithm of Madigan and York (1995).

4. Bootstrap aggregation or bagging (**BAG**): Bagging involves generating a large number of bootstrap samples from the original data, calculating least squares estimates, conducting two-sided t-tests on each parameter estimate, generating forecasts and averaging forecasts across bootstrap samples. The exact settings and critical values used are identical to the ones proposed in Ribeiro (2016, pages 6-7).

5. Five factors (**DFM5**): The DFM-5 forecast uses the first five principle components as predictors, with coefficients estimated by OLS without shrinkage. This is a parsimonious specification and can be thought of as a naive approach to shrinkage that, in practice, may perform better than the state-of-the-art methods outlined above.

6. Full model (**OLS**): As an indication of how well shrinkage methods perform, I quote results from the model with all 50 factors included estimated with unrestricted least squares.

7. Minimal model (**ARp**): This is a the model with zero factors and only a single lag of the dependent variable, estimated by OLS. I do not explicitly quote results for this model, rather I use it as a reference point for other methods (i.e. all results are relative to the AR(1) model).

I use the second half of the sample to evaluate forecasts from the regression model using all estimation methods outlined above. The first estimation period is 1959Q1-1985Q4, and I forecast $h = 1, 2, 4$ and 8 steps ahead. Estimation and forecasting is recursive, adding each quarter one observation to the initial estimation sample, until the whole sample is exhausted. That is, the last estimation period is 1959Q1 - (2015Q2-h). Additional results based on rolling forecasts are presented in the Appendix. Given that Bayesian and non-Bayesian estimation methods are compared, I focus on point estimation only, following closely Stock and Watson (2012). Results are typically evaluated by taking two standard distance functions of forecast errors, the rectilinear (ℓ_1-norm) and Euclidean (ℓ_2-norm), leading to the mean absolute forecast error (MAFE) and the root mean square forecast error (RMFSE).

4.2 Main Results

This section presents the main results of this paper for all 222 series and for the case of recursive forecasts. Additional results for rolling forecasts and for selected series are discussed in the next subsection, and additional tables can be found in the Appendix.

Table 2 reports the distributions of relative root mean squared errors (RMSEs) for the various forecasting methods. The rows represent percentiles of the distribution of RMSEs over the 222 series for the eight forecasting methods, where all RMSEs are relative to the AR(1) benchmark. Values less than one signify improvement over the benchmark, while values larger than one suggest that the benchmark performs well. First note that the general results are in-

line with the findings of Stock and Watson (2012): All shrinkage or parsimonious methods (that is, excluding OLS using all 50 factors) improve over the AR(1) benchmark. The improvements in some cases are not as large as the ones that Stock and Watson find in their respective Table 2, but note that they use an AR(4) benchmark (which, in general, is inferior to the AR(1) for this kind of data). Additionally, it can also be seen that the forecast improvement from using shrinkage estimators over the naive, but parsimonious, DFM5 approach, is modest. The only consistent pattern that seems to emerge is that in the 95% percentile, the shrinkage methods do not generate as large RMSFEs as the DFM5. For example, for $h = 8$ DFM5 has RMSFE of the order of 32% worse than the benchmark AR(1), while the vast majority of shrinkage estimators provide reduction in forecast accuracy of 10%-20%.

These important observations aside, the main research question in this paper is how GAMP-based algorithms perform compared to other shrinkage algorithms. The answer to that question is strikingly straightforward to respond to based on evidence in Table 2. In particular GAMP-SBL seems to perform extremely well, but also GAMP-SNS follows closely. In terms of the median SBL is better than DFM5 in three out of four forecast horizons, while SNS is performing satisfactorily, meaning that it improves AR(1) forecasts and at the same time is comparable to the performance of the remaining shrinkage methods. In terms of the 5^{th} percentile, all shrinkage methods and the DFM5 are comparable. However, when looking at the 95^{th} percentiles, SBL and SNS generate lower RMSFEs compared to DFM5.

The reason why GAMP-SBL and GAMP-SNS perform consistently well at all forecast horizons, while alternative shrinkage methods might perform well in some cases (e.g. for $h = 1$) but not so well in others (e.g. for $h = 8$ all other shrinkage methods, apart from the DFM5, perform worse than the AR(1) in the median), is simply the fact that GAMP-based algorithms do not require tuning of prior quantities, since all hyperpriors are estimated automatically using the EM updates [9]. So when correlations between the predictors and the variable of interest, y_{t+h}, change as the horizon h changes, GAMP-based algorithms will adapt automatically to

[9]This tuning requirement also holds for bagging. Even though this algorithm is not based on a Bayesian prior, it requires selection of a critical value to serve as a threshold for retaining predictors during all bootstrap iterations.

the new setting. Given the simplicity and tractability of GAMP, one can think of even further enhancements, for example run multiple instances of GAMP using different initial parameter guesses in order to eliminate the effect of initial conditions and increase its precision even further. MCMC-based algorithms may also suffer from the effect of initial conditions, but in many cases it may be computationally impractical to run multiple instances of an MCMC algorithm.

Table 3 summarizes the results of the previous Table by considering the weighted mean square forecast error measure (WMSFE) used in Christoffersen and Diebold (1998). This is a multivariate measure, where the scale of each of the 222 variables is used in order to construct a weighted sum of all univariate MSFEs. Results are also standardized to be relative to the AR(1) WMSFE. In this table we also see that using this criterion most methods do not perform better than the AR(1), but SBL and SNS are better, especially SBL for $h = 3$ where improvements look substantial. Table 4 explores whether forecasts from the competing methods differ from each other. Specifically, this Table presents two measures of similarity of the performance of one-step ahead forecasts, namely the correlation (over series) among RMSEs, relative the AR(1) forecasts, and the mean absolute difference of these relative RMSEs. All shrinkage forecasts are highly correlated, while DFM5 and OLS are less correlated with other methods and with each other. Nevertheless, the parsimonious DFM5 generates mean absolute differences which are similar to ones by the shrinkage methods, while the non-parsimonious OLS fails to a large extent to perform well. This observation suggests two things. First, the two GAMP-based algorithms generate RMSFEs in accordance with other shrinkage methods. Second, even though the first few principal components typically explain most of the variability in a dataset (such as the one used here to extract factors), the shrinkage methods might not be selecting consistently the first few components. If that was the case, their RMSFEs would not have such low correlation with the DFM5 forecasts. This low correlation might suggest that shrinkage methods retain possibly the first few components as well as a mix of less informative components. Finally, Table 5 shows the hit rates in terms of mean absolute and root mean square forecast errors of the eight competing methods. These hit rates simply measure the

percentage of times, over the 222 series, that each method had achieved the lowest MAFE or RMSFE. Again, we can see that this criterion also confirms the excellent performance of SBL and particularly SNS compared to all other methods. It is noteworthy that the GAMP-based methods retain over all forecast horizons their good percentages of being the best performing methods, while SSVS has gradually decreasing hit-rates while DFM5 consistently improves over the forecast horizons.

Table 2: Distributions of Relative Root Mean Squared Errors (RMSE), Relative to the AR(1) Forecast, by Forecasting Method, all 222 series

				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
0.075	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

Notes: Entries are percentiles of distributions of relative RMSFEs over the 222 variables being forecasted, by series, at the 1-, 2-, 4- and 8-quarter ahead forecast horizon. RMSFEs are relative to the AR(1) forecast RMSFE, and are computed using an expanding window of in-sample observations (recursive). All forecasts are direct.

Table 3: Multivariate weighted mean squared forecast error, for each estimator and forecast horizon, all 222 series

	$h=1$	$h=2$	$h=4$	$h=8$
SBL	0.986	0.980	0.905	0.992
SNS	0.998	0.996	1.032	0.986
LASSO	1.044	1.065	1.106	1.164
SSVS	1.000	1.002	1.076	1.179
BMA	1.034	1.045	1.058	1.133
BAG	1.019	1.046	1.155	0.987
DFM5	0.993	1.077	1.066	1.142
OLS	1.043	1.029	1.126	1.035

Notes: Entries in this table report the multivariate weighted mean squared forecast error of each method relative to the AR(1); see Christoffersen and Diebold (1998) for more details.

Table 4: Two Measures of Similarity of Forecast Performance, $h=1$: Correlation (lower left) and Mean Absolute Difference of Forecasts (upper right), all 222 series

	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
SBL		0.036	0.041	0.029	0.023	0.039	0.046	0.229
SNS	0.87		0.046	0.025	0.037	0.044	0.058	0.229
LASSO	0.90	0.86		0.047	0.038	0.038	0.052	0.202
SSVS	0.85	0.88	0.82		0.020	0.044	0.046	0.236
BMA	0.90	0.84	0.87	0.96		0.035	0.043	0.225
BAG	0.88	0.80	0.89	0.83	0.89		0.049	0.209
DFM5	0.24	0.06	0.29	0.17	0.28	0.47		0.209
OLS	0.44	0.32	0.43	0.48	0.51	0.54	0.59	

Notes: Entries below the diagonal are the correlation between the RMSFEs for the row/column forecasting methods, computed over the 222 series being forecasted. Entries above the diagonal are the mean absolute difference between the row/column method RMSFEs, averaged across series.

Table 5: Hit-rates of the eight estimators, all 222 series

				Horizon $h=1$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	14.41	19.37	4.96	23.42	6.31	13.06	14.41	4.05
% of lowest RMSFE	15.32	22.07	5.41	16.67	8.56	13.51	14.41	4.05
				Horizon $h=2$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	12.61	31.53	4.05	10.36	3.15	20.72	13.51	4.05
% of lowest RMSFE	11.71	26.13	3.15	11.26	6.76	15.77	20.72	4.50
				Horizon $h=4$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	11.71	27.93	3.60	10.36	4.96	18.02	20.27	3.15
% of lowest RMSFE	11.26	26.13	5.41	10.36	4.96	13.96	23.87	4.05
				Horizon $h=8$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	8.11	29.73	5.86	9.01	3.60	12.61	27.93	3.15
% of lowest RMSFE	9.91	24.32	7.21	7.21	1.80	13.06	33.78	2.70

Note: This table shows the proportion of times (over the 222 series being forecasted) that each estimator achieved the lowest value of the MAFE and RMSFE statistics.

4.3 Further results and discussion

In the Appendix I provide additional results that help solidify the conclusions drawn from the main results. First, I consider the results only on 14 series that are possibly the most important indicators that macroeconomists and analysts look at. These are real GDP, personal consumption expenditures, industrial production (total), employment, unemployment rate, GDP price deflator, consumer price index (total), producer price index (commodities), federal funds rate, 10-year Treasury bond rate, real M1 money stock, GBP/USD exchange rate, and S&P500 stock prices. There are, of course, several leading indicators that one could also follow (hours worked, consumer confidence, mortgage rates etc), nevertheless these 14 series roughly describe the variables of interest (target variables) for policy-makers and central banks. The main argument of doing this exercise is that, when evaluating 222 series, it might be the case that GAMP is performing really well on series which are higher level disaggregates (and possibly not so interesting), while performing poorly on variables that matter. Tables C.1 to

C4 show that this is not the case.

Qualitatively similar conclusions can be drawn when considering rolling forecasts, that is, forecasts with in-sample estimation of coefficients in a fixed window of observations. Such an approach takes into account evident structural breaks in macroeconomic time series (see Bauwens, Koop, Korobilis and Rombouts, 2015). I follow Stock and Watson (2012) and consider a rolling window of 100 observations, using the same evaluation period for all forecasts. Doing so, means that all shrinkage estimators should "work harder" to find suitable restrictions on regression coefficients and save valuable degrees of freedom, and for that reason parsimonious approaches (the AR(1) and DFM5) are expected to have a possible advantage a-priori. Nevertheless, results are favourable for all shrinkage estimators with SBL and SNS leading the course.

Considering rolling forecasts is a simple approach to control for structural breaks, but choice of the rolling window is subjective. One can consider, instead, specifications that explicitly account for changing volatility, which has been shown to be an extremely important feature of macroeconomic data; see Clark and Ravazzollo (2015). In the Technical Appendix I show different ways to combine the GAMP shrinkage estimators with stochastic volatility. The results from an empirical illustration can be seen in Table B1, where extensions of GAMP that account for stochastic volatility visibly improve forecast performance compared to variants with constant volatility. Finally, note that other extensions of GAMP are readily available. In a vector autoregressive (VAR) setting for example, if we consider the decomposition in Carrierro, Clark and Marcellino (2016) that allows for equation-by-equation estimation of the VAR, then application of GAMP is trivial. Given the recent interest in estimating large VARs using approximations or machine learning methods (see Koop, Korobilis and Pettenuzzo, 2017, and references therein), GAMP can provide an additional estimation tool for this class of models. Similar adaptations to factor models and other popular multivariate specifications are also possible, and they would be an interesting direction for future research in applied econometrics.

5 Conclusions

This paper evaluates a new methodology for performing Bayesian inference in high-dimensional regression models. The proposed Generalized Approximate Message Passing (GAMP) is a fast algorithm for approximating iteratively the first two moments of the marginal posterior distribution of high-dimensional coefficients. It is established how effortlessly GAMP can be combined with two popular classes of shrinkage priors, and extensive evaluations in synthetic and real data demonstrate the advantages of using GAMP. In particular, GAMP is at least as precise as a wide class of alternative shrinkage methods, at a fraction of the computing time and with minimal user input. While the kind of application considered in this paper is typical in macroeconomics, it is far from being a truly "Big Data" application. Nevertheless, GAMP will accommodate regression models with thousands or, possibly, millions of predictors before it hits a computational bottleneck. This paper, thus, attempts to make the point that GAMP should become a valuable tool for applied economists and policy-makers who wish to monitor high-dimensional datasets, and that, despite its approximate nature, this fast algorithm is an investment for future generations of macroeconomists who might be faced with a daunting amount of available information to process.

References

[1] Barber, D. (2012). Bayesian reasoning and machine learning. Cambridge University Press: New York.

[2] Baron, D., Sarvotham, S. and Baraniuk, R. G. (2010). Bayesian compressive sensing via belief propagation. *IEEE Transactions in Signal Processing*, 58, 269-280.

[3] Bauwens, L., Koop, G., Korobilis, D. and Rombouts, J. V. K. (2015). The contribution of structural break models to forecasting macroeconomic series. *Journal of Applied Econometrics*, 30(4), 596-620.

[4] Bayati, M. and Montanari, A. (2011). The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57(2), 764-785.

[5] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24 (2), 123-140.

[6] Carriero, A., Clark, T. E. and Marcellino, M. (2016). Large vector autoregressions with stochastic volatility and flexible priors. Working paper 16-17, Federal Reserve Bank of Cleveland.

[7] Christoffersen, P. and Diebold, F. (1998). Cointegration and long-horizon forecasting. *Journal of Business and Economic Statistics*, 16(4), 450-458.

[8] Clark, T. E. (2011). Real-time density forecasts from Bayesian vector autoregressions with stochastic volatility. *Journal of Business and Economic Statistics*, 29(3), 327-341.

[9] Clark, T. E. and Ravazzolo, F. (2015). Macroeconomic forecasting performance under alternative specifications of timevarying volatility. *Journal of Applied Econometrics*, 30(4), 551-575.

[10] Donoho, D. L., Maleki, A. and Montanari, A. (2009). Message passing algorithms for compressed sensing. *Proceedings of National Academy of Sciences*, 106(45), 18914-18919.

[11] Donoho, D. L., Maleki, A. and Montanari, A. (2011). How to design message passing algorithms for compressed sensing. Unpublished manuscript, available at http://www.ece.rice.edu/ mam15/bpist.pdf.

[12] Fernandez, C., E. Ley, and Steel, M. (2001a). Model uncertainty in cross-country growth regressions. *Journal of Applied Econometrics*, 16, 563576.

[13] Fernandez, C., E. Ley, and Steel, M. (2001b). Benchmark priors for Bayesian model averaging. *Journal of Econometrics*, 100, 381-427.

[14] Freeman, W. T., Pasztor, E. C., and Carmichael, O. T. (2000). Learning low-level vision. *International Journal of Computer Vision*, 40, 2547.

[15] George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423), 881-889.

[16] Griffin, J. E. and Brown, P. J. (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5(1), 171-188.

[17] Ji, S., Xue, Y. and Carin, L. (2008). Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6), 2346-2356.

[18] Kamilov, U. S., Rangan, S., Fletcher, A. K. and Unser, M. (2014). Approximate message passing with consistent parameter estimation and applications to sparse learning. *IEEE Transactions on Information Theory*, 60(5), 2969-2985.

[19] Koop, G. and Korobilis, D. (2012). Forecasting inflation using dynamic model averaging. *International Economic Review*, 53, 867-886

[20] Koop, G., Korobilis, D. and Pettenuzzo, D. (2017). Bayesian compressed vector autoregressions. forthcoming *Journal of Econometrics*.

[21] Korobilis, D. (2013). Hierarchical shrinkage priors for dynamic regressions with many predictors. *International Journal of Forecasting*, 29, 43-59.

[22] Kschischang, F. R., Frey, B. J. and Loeliger, H. A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2), 498-519.

[23] Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review*, 63, 215-232.

[24] McEliece, R. J., MacKay, D. J. C. and Cheng, J. (1998). Turbo decoding as an instance of Pearls belief propagation algorithm. *IEEE Journal on Selected Areas in Communications*, 16, 140-152.

[25] Park, T. and Casella, G. (2008). The Bayesian lasso. *Journal of the American Statistical Association*, 103(482), 681-686.

[26] Pearl, J. (1982). Reverend Bayes on inference engines: A distributed hierarchical approach. AAAI-82: Pittsburgh, PA. Second National Conference on Artificial Intelligence. Menlo Park, California: AAAI Press, 133-136.

[27] Rangan, S. (2011). Generalized approximate message passing for estimation with random linear mixing. *IEEE International Symposium on Information Theory*, 2174-2178.

[28] Rangan, S., Schniter, P., Riegler, E., Fletcher, A. K. and Cevher, V. (2016). Fixed points of generalized approximate message passing with arbitrary matrices. arXiv:1301.6295v4

[29] Ribeiro, P. (2016). Revealing Exchange Rate Fundamentals by Bootstrap. Available at SSRN: https://ssrn.com/abstract=2839259.

[30] Stock, J. H. and Watson, M. W. (2006). Macroeconomic Forecasting Using Many Predictors. Handbook of Economic Forecasting, Graham Elliott, Clive Granger, Allan Timmerman (eds.), North Holland, 2006.

[31] Stock, J. H. and Watson, M. W. (2012). Generalized shrinkage methods for forecasting using many predictors. *Journal of Business and Economic Statistics*, 30(4), 481-493.

[32] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58, 267-288.

[33] Vila, J. P. and Schniter, P. (2011). Expectation-maximization Bernoulli-Gaussian approximate message passing. *Conference on Signals, Systems, and Computing (ASILOMAR)*, Pacific Grove, CA, USA, 799-803.

[34] Vila, J. P. and Schniter, P. (2013). Expectation-maximization gaussian-mixture approximate message passing. *IEEE Transactions on Signal Processing*, 61(19), 4658-4672.

[35] West, M. and Harrison, P. J. (1997). *Bayesian Forecasting and Dynamic Models* (2nd ed.). Springer: New York.

[36] Ziniel, J. and Schniter, P. (2013). Dynamic compressive sensing of time-varying signals via approximate message passing. *IEEE Transactions on Signal Processing*, 61(21), 5270-5284.

[37] Ziniel, J., Potter, C. and Schniter, P. (2010). Tracking and smoothing of time-varying sparse signals via approximate belief propagation. *Procedures of Forty-fouth Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA.

[38] Zou, X., Li, F., Fang, J. and Li, H. (2016). Computationally efficient sparse Bayesian learning via generalized approximate message passing. IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), 1-4.

A Data Appendix

All series were downloaded from Michael McCracken's FRED-QD database (https://research.stlouisfed.org/econ/mccracken/fred-databases/) in March 2017 and cover the quarters 1959Q1 to 2015Q3. Out of the original 257 series, only 222 have been retained in order to ensure a balanced panel of observations. All series are seasonally adjusted and all variables are transformed to be approximately stationary and the transformation codes for each variable appear in the column 'T' on the table below.

In particular, if $w_{i,t}$ is the original un-transformed series in levels, when the series is used as a predictor via factors (R.H.S. of the regression model) the transformation codes are: 1 - no transformation (levels), $x_{i,t} = w_{i,t}$; 2 - first difference, $x_{i,t} = w_{i,t} - w_{i,t-1}$; 3- second difference, $x_{i,t} = \Delta w_{i,t} - \Delta w_{i,t-1}$ 4 - logarithm, $x_{i,t} = \log w_{i,t}$; 5 - first difference of logarithm, $x_{i,t} = \log w_{i,t} - \log w_{i,t-1}$; 6 - second difference of logarithm, $x_{i,t} = \Delta \log w_{i,t} - \Delta \log w_{i,t-1}$.

When the series is used as the variable to be predicted (L.H.S. of the regression model) the transformation codes are: 1 - no transformation (levels), $y_{i,t+h} = w_{i,t+h}$; 2 - first difference, $y_{i,t+h} = w_{i,t+h} - w_{i,t}$; 3- second difference, $y_{i,t+h} = \frac{1}{h}\Delta^h w_{i,t+h} - \Delta w_{i,t}$ 4 - logarithm, $y_{i,t+h} = \log w_{i,t+h}$; 5 - first difference of logarithm, $y_{i,t+h} = \log w_{i,t+h} - \log w_{i,t}$; 6 - second difference of logarithm, $y_{i,t+h} = \frac{1}{h}\Delta^h \log w_{i,t+h} - \Delta \log w_{i,t}$. In the transformations above, I define $\Delta w_t = w_t - w_{t-1}$ and $\Delta^h w_{t+h} = w_{t+h} - w_t$.

From the 222 series, 92 are higher level aggregates and do not add information when extracting principal components. These series are indicated with a 0 in column 'F' of the table below, and only the remaining 130 series are used for extracting factors.

Table A1: Quarterly US macro dataset (based on FREDQD)

No	Mnemonic	F	T	Long description
1	GDPC96	0	5	Real Gross Domestic Product, 3 Decimal (Billions of Chained 2009 Dollars)
2	PCECC96	0	5	Consumption Real Personal Consumption Expenditures (Billions of Chained 2009 Dollars)
3	PCDGx	1	5	Real personal consumption expenditures: Durable goods (Billions of Chained 2009 Dollars), deflated using PCE
4	PCESVx	1	5	Real Personal Consumption Expenditures: Services (Billions of 2009 Dollars), deflated using PCE
5	PCNDx	1	5	Real Personal Consumption Expenditures: Nondurable Goods (Billions of 2009 Dollars), deflated using PCE
6	GPDIC96	0	5	Real Gross Private Domestic Investment, 3 decimal (Billions of Chained 2009 Dollars)
7	FPIx	0	5	Real private fixed investment (Billions of Chained 2009 Dollars), deflated using PCE
8	Y033RC1Q027SBEAx	1	5	Real Gross Private Domestic Fixed Investment: Nonresidential: Equipment (Billions of Chained 2009 Dollars), deflated using PCE
9	PNFIx	1	5	Real private fixed investment: Nonresidential (Billions of Chained 2009 Dollars), deflated using PCE
10	PRFIx	1	5	Real private fixed investment: Residential (Billions of Chained 2009 Dollars), deflated using PCE
11	A014RE1Q156NBEA	1	1	Inventories Shares of gross domestic product: Gross private domestic investment: Change in private inventories (Percent)
12	GCEC96	0	5	Real Government Consumption Expenditures & Gross Investment (Billions of Chained 2009 Dollars)
13	A823RL1Q225SBEA	1	1	Real Government Consumption Expenditures and Gross Investment: Federal (Percent Change from Preceding Period)
14	FGRECPTx	1	5	Real Federal Government Current Receipts (Billions of Chained 2009 Dollars), deflated using PCE
15	SLCEx	1	5	Real government state and local consumption expenditures (Billions of Chained 2009 Dollars), deflated using PCE
16	EXPGSC96	1	5	Real Exports of Goods & Services, 3 Decimal (Billions of Chained 2009 Dollars)
17	IMPGSC96	1	5	Real Imports of Goods & Services, 3 Decimal (Billions of Chained 2009 Dollars)
18	DPIC96	0	5	Real Disposable Personal Income (Billions of Chained 2009 Dollars)
19	OUTNFB	0	5	Real Disposable Personal Income (Billions of Chained 2009 Dollars)
20	OUTBS	0	5	Business Sector: Real Output (Index 2009=100)
21	INDPRO	0	5	Industrial Production Index (Index 2012=100)
22	IPFINAL	0	5	Industrial Production: Final Products (Market Group) (Index 2012=100)
23	IPCONGD	0	5	Industrial Production: Consumer Goods (Index 2012=100)
24	IPMAT	0	5	Industrial Production: Materials (Index 2012=100)
25	IPDMAT	1	5	Industrial Production: Durable Materials (Index 2012=100)
26	IPNMAT	1	5	Industrial Production: Nondurable Materials (Index 2012=100)
27	IPDCONGD	1	5	Industrial Production: Durable Consumer Goods (Index 2012=100)
28	IPB51110SQ	1	5	Industrial Production: Durable Goods: Automotive products (Index 2012=100)

Table A1 (continued)

29	IPNCONGD	1	5	Industrial Production: Durable Goods: Automotive products (Index 2012=100)
30	IPBUSEQ	1	5	Industrial Production: Business Equipment (Index 2012=100)
31	IPB51220SQ	1	5	Industrial Production: Consumer energy products (Index 2012=100)
32	CUMFNS	1	1	Capacity Utilization: Manufacturing (SIC) (Percent of Capacity)
33	PAYEMS	0	5	All Employees: Total nonfarm (Thousands of Persons)
34	USPRIV	0	5	All Employees: Total Private Industries (Thousands of Persons)
35	MANEMP	0	5	All Employees: Manufacturing (Thousands of Persons)
36	SRVPRD	0	5	All Employees: Service-Providing Industries (Thousands of Persons)
37	USGOOD	0	5	All Employees: Goods-Producing Industries (Thousands of Persons)
38	DMANEMP	1	5	All Employees: Durable goods (Thousands of Persons)
39	NDMANEMP	0	5	All Employees: Nondurable goods (Thousands of Persons)
40	USCONS	1	5	All Employees: Construction (Thousands of Persons)
41	USEHS	1	5	All Employees: Education & Health Services (Thousands of Persons)
42	USFIRE	1	5	All Employees: Education & Health Services (Thousands of Persons)
43	USINFO	1	5	All Employees: Information Services (Thousands of Persons)
44	USPBS	1	5	All Employees: Professional & Business Services (Thousands of Persons)
45	USLAH	1	5	All Employees: Leisure & Hospitality (Thousands of Persons)
46	USSERV	1	5	All Employees: Other Services (Thousands of Persons)
47	USMINE	1	5	All Employees: Mining and logging (Thousands of Persons)
48	USTPU	1	5	All Employees: Trade, Transportation & Utilities (Thousands of Persons)
49	USGOVT	0	5	All Employees: Government (Thousands of Persons)
50	USTRADE	1	5	All Employees: Retail Trade (Thousands of Persons)
51	USWTRADE	1	5	All Employees: Wholesale Trade (Thousands of Persons)
52	CES9091000001	1	5	All Employees: Government: Federal (Thousands of Persons)
53	CES9092000001	1	5	All Employees: Government: State Government (Thousands of Persons)
54	CES9093000001	1	5	All Employees: Government: Local Government (Thousands of Persons)
55	CE16OV	0	5	Civilian Employment (Thousands of Persons)
56	CIVPART	0	2	Civilian Labor Force Participation Rate (Percent)
57	UNRATE	0	2	Civilian Unemployment Rate (Percent)
58	UNRATESTx	0	2	Unemployment Rate less than 27 weeks (Percent)
59	UNRATELTx	0	2	Unemployment Rate for more than 27 weeks (Percent)
60	LNS14000012	1	2	Unemployment Rate - 16 to 19 years (Percent)
61	LNS14000025	1	2	Unemployment Rate - 20 years and over, Men (Percent)
62	LNS14000026	1	2	Unemployment Rate - 20 years and over, Women (Percent)
63	UEMPLT5	1	5	Number of Civilians Unemployed - Less Than 5 Weeks (Thousands of Persons)
64	UEMP5TO14	1	5	Number of Civilians Unemployed for 5 to 14 Weeks (Thousands of Persons)
65	UEMP15T26	1	5	Number of Civilians Unemployed for 15 to 26 Weeks (Thousands of Persons)
66	UEMP27OV	1	5	Number of Civilians Unemployed for 27 Weeks and Over (Thousands of Persons)
67	LNS12032194	1	5	Employment Level - Part-Time for Economic Reasons, All Industries (Thousands of Persons)
68	HOABS	0	5	Business Sector: Hours of All Persons (Index 2009=100)
69	HOANBS	0	5	Nonfarm Business Sector: Hours of All Persons (Index 2009=100)
70	AWHMAN	1	1	Average Weekly Hours of Production and Nonsupervisory Employees: Manufacturing (Hours)
71	AWOTMAN	1	2	Average Weekly Hours Of Production And Nonsupervisory Employees: Total private (Hours)
72	HWIx	0	1	Help-Wanted Index
73	HOUST	0	5	Housing Starts: Total: New Privately Owned Housing Units Started (Thousands of Units)
74	HOUST5F	0	5	Housing Starts: Total: New Privately Owned Housing Units Started (Thousands of Units)
75	PERMIT	1	5	New Private Housing Units Authorized by Building Permits (Thousands of Units)
76	HOUSTMW	1	5	Housing Starts in Midwest Census Region (Thousands of Units)
77	HOUSTNE	1	5	Housing Starts in Northeast Census Region (Thousands of Units)
78	HOUSTS	1	5	Housing Starts in South Census Region (Thousands of Units)
79	HOUSTW	1	5	Housing Starts in West Census Region (Thousands of Units)
80	CMRMTSPLx	0	5	Real Manufacturing and Trade Industries Sales (Millions of Chained 2009 Dollars)
81	RSAFSx	1	5	Real Retail and Food Services Sales (Millions of Chained 2009 Dollars), deflated by Core PCE
82	AMDMNOx	1	5	Real Manufacturers New Orders: Durable Goods (Millions of 2009 Dollars), deflated by Core PCE
83	AMDMUOx	1	5	Real Manufacturers Unfilled Orders for Durable Goods (Million of 2009 Dollars), deflated by Core PCE
84	NAPMSDI	1	1	ISM Manufacturing: Supplier Deliveries Index (lin)
85	PCECTPI	0	6	Personal Consumption Expenditures: Chain-type Price Index (Index 2009=100)
86	PCEPILFE	0	6	Personal Consumption Expenditures Excluding Food and Energy (Index 2009=100)
87	GDPCTPI	0	6	Gross Domestic Product: Chain-type Price Index (Index 2009=100)
88	GPDICTPI	1	6	Gross Private Domestic Investment: Chain-type Price Index (Index 2009=100)
89	IPDBS	1	6	Business Sector: Implicit Price Deflator (Index 2009=100)

Table A1 (continued)

90	DGDSRG3Q086SBEA	0	6	Goods Personal consumption expenditures: Goods (chain-type price index)
91	DDURRG3Q086SBEA	0	6	Personal consumption expenditures: Durable goods (chain-type price index)
92	DSERRG3Q086SBEA	0	6	Personal consumption expenditures: Services (chain-type price index)
93	DNDGRG3Q086SBEA	0	6	Personal consumption expenditures: Nondurable goods (chain-type price index)
94	DHCERG3Q086SBEA	0	6	Personal consumption expenditures: Nondurable goods (chain-type price index)
95	DMOTRG3Q086SBEA	1	6	Personal consumption expenditures: Durable goods: Motor vehicles and parts (chain-type price index)
96	DFDHRG3Q086SBEA	1	6	Personal consumption expenditures: Durable goods: Furnishings and durable household equipment (chain-type price index)
97	DREQRG3Q086SBEA	1	6	Personal consumption expenditures: Durable goods: Recreational goods and vehicles (chain-type price index)
98	DODGRG3Q086SBEA	1	6	Personal consumption expenditures: Durable goods: Other durable goods (chain-type price index)
99	DFXARG3Q086SBEA	1	6	Personal consumption expenditures: Nondurable goods: Food and beverages purchased for off-premises consumption (chain-type price index)
100	DCLORG3Q086SBEA	1	6	Personal consumption expenditures: Nondurable goods: Clothing and footwear (chain-type price index)
101	DGOERG3Q086SBEA	1	6	Personal consumption expenditures: Nondurable goods: Gasoline and other energy goods (chain-type price index)
102	DONGRG3Q086SBEA	1	6	Personal consumption expenditures: Nondurable goods: Other nondurable goods (chain-type price index)
103	DHUTRG3Q086SBEA	1	6	Personal consumption expenditures: Services: Housing and utilities (chain-type price index)
104	DHLCRG3Q086SBEA	1	6	Personal consumption expenditures: Services: Health care (chain-type price index)
105	DTRSRG3Q086SBEA	1	6	Personal consumption expenditures: Transportation services (chain-type price index)
106	DRCARG3Q086SBEA	1	6	Personal consumption expenditures: Recreation services (chain-type price index)
107	DFSARG3Q086SBEA	1	6	Personal consumption expenditures: Services: Food services and accommodations (chain-type price index)
108	DIFSRG3Q086SBEA	1	6	Personal consumption expenditures: Financial services and insurance (chain-type price index)
109	DOTSRG3Q086SBEA	1	6	Personal consumption expenditures: Other services (chain-type price index)
110	CPIAUCSL	0	6	Consumer Price Index for All Urban Consumers: All Items (Index 1982-84=100)
111	CPILFESL	0	6	Consumer Price Index for All Urban Consumers: All Items Less Food & Energy (Index 1982-84=100)
112	PPIFGS	0	6	Producer Price Index by Commodity for Finished Goods (Index 1982=100)
113	PPIACO	0	6	Producer Price Index for All Commodities (Index 1982=100)
114	PPIFCG	1	6	Producer Price Index by Commodity for Finished Consumer Goods (Index 1982=100)
115	PPIFCF	1	6	Producer Price Index by Commodity for Finished Consumer Foods (Index 1982=100)
116	PPIIDC	1	6	Producer Price Index by Commodity Industrial Commodities (Index 1982=100)
117	PPIITM	1	6	Producer Price Index by Commodity Intermediate Materials: Supplies & Components (Index 1982=100)
118	NAPMPRI	1	1	ISM Manufacturing: Prices Index (Index)
119	WPU0561	1	5	Producer Price Index by Commodity for Fuels and Related Products and Power: Crude Petroleum (Domestic Production) (Index 1982=100)
120	OILPRICEx	0	5	Real Crude Oil Prices: West Texas Intermediate (WTI) - Cushing, Oklahoma (2009 Dollars per Barrel), deflated by Core PCE
121	CES2000000008x	0	5	Real Average Hourly Earnings of Production and Nonsupervisory Employees: Construction (2009 Dollars per Hour), deflated by Core PCE
122	CES3000000008x	0	5	Real Average Hourly Earnings of Production and Nonsupervisory Employees: Manufacturing (2009 Dollars per Hour), deflated by Core PCE
123	COMPRNFB	1	5	Manufacturing Sector: Real Compensation Per Hour (Index 2009=100)
124	RCPHBS	1	5	Business Sector: Real Compensation Per Hour (Index 2009=100)
125	OPHNFB	1	5	Nonfarm Business Sector: Real Output Per Hour of All Persons (Index 2009=100)
126	OPHPBS	0	5	Business Sector: Real Output Per Hour of All Persons (Index 2009=100)
127	ULCBS	0	5	Business Sector: Unit Labor Cost (Index 2009=100)
128	ULCNFB	1	5	Nonfarm Business Sector: Unit Labor Cost (Index 2009=100)
129	UNLPNBS	1	5	Nonfarm Business Sector: Unit Nonlabor Payments (Index 2009=100)
130	FEDFUNDS	1	2	Effective Federal Funds Rate (Percent)
131	TB3MS	1	2	3-Month Treasury Bill: Secondary Market Rate (Percent)
132	TB6MS	0	2	6-Month Treasury Bill: Secondary Market Rate (Percent)
133	GS1	0	2	1-Year Treasury Constant Maturity Rate (Percent)
134	GS10	0	2	10-Year Treasury Constant Maturity Rate (Percent)
135	AAA	0	2	Moodys Seasoned Aaa Corporate Bond Yield (Percent)
136	BAA	0	2	Moodys Seasoned Baa Corporate Bond Yield (Percent)
137	BAA10YM	1	1	Moodys Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity (Percent)
138	TB6M3Mx	1	1	6-Month Treasury Bill Minus 3-Month Treasury Bill, secondary market (Percent)
139	GS1TB3Mx	1	1	1-Year Treasury Constant Maturity Minus 3-Month Treasury Bill, secondary market (Percent)
140	GS10TB3Mx	1	1	10-Year Treasury Constant Maturity Minus 3-Month Treasury Bill, secondary market (Percent)
141	CPF3MTB3Mx	1	1	3-Month Commercial Paper Minus 3-Month Treasury Bill, secondary market (Percent)
142	AMBSLREALx	1	5	St. Louis Adjusted Monetary Base (Billions of 1982-84 Dollars), deflated by CPI
143	M1REALx	1	5	Real M1 Money Stock (Billions of 1982-84 Dollars), deflated by CPI
144	M2REALx	1	5	Real M2 Money Stock (Billions of 1982-84 Dollars), deflated by CPI
145	MZMREALx	1	5	Real MZM Money Stock (Billions of 1982-84 Dollars), deflated by CPI
146	BUSLOANSx	1	5	Real Commercial and Industrial Loans, All Commercial Banks (Billions of 2009 U.S. Dollars), deflated by Core PCE
147	CONSUMERx	1	5	Consumer Loans at All Commercial Banks (Billions of 2009 U.S. Dollars), deflated by Core PCE
148	NONREVSLx	1	5	Total Real Nonrevolving Credit Owned and Securitized, Outstanding (Billions of Dollars), deflated by Core PCE
149	REALLNx	1	5	Real Real-Estate Loans, All Commercial Banks (Billions of 2009 U.S. Dollars), deflated by Core PCE
150	TOTALSLx	1	5	Total Consumer Credit Outstanding, deflated by Core PCE

Table A1 (continued)

151	TABSHNOx	1	5	Real Total Assets of Households and Nonprofit Organizations (Billions of 2009 Dollars), deflated by Core PCE
152	TLBSHNOx	1	5	Real Total Liabilities of Households and Nonprofit Organizations (Billions of 2009 Dollars), deflated by Core PCE
153	LIABPIx	0	5	Liabilities of Households and Nonprofit Organizations Relative to Personal Disposable Income (Percent)
154	TNWBSHNOx	1	5	Real Net Worth of Households and Nonprofit Organizations (Billions of 2009 Dollars), deflated by Core PCE
155	NWPIx	0	1	Net Worth of Households and Nonprofit Organizations Relative to Disposable Personal Income (Percent)
156	TARESAx	1	5	Real Assets of Households and Nonprofit Organizations excluding Real Estate Assets (Billions of 2009 Dollars), deflated by Core PCE
157	HNOREMQ027Sx	1	5	Real Real-Estate Assets of Households and Nonprofit Organizations (Billions of 2009 Dollars), deflated by Core PCE
158	TFAABSHNOx	1	5	Real Total Financial Assets of Households and Nonprofit Organizations (Billions of 2009 Dollars), deflated by Core PCE
159	EXSZUSx	1	5	Switzerland / U.S. Foreign Exchange Rate
160	EXJPUSx	1	5	Japan / U.S. Foreign Exchange Rate
161	EXUSUKx	1	5	U.S. / U.K. Foreign Exchange Rate
162	EXCAUSx	1	5	Canada / U.S. Foreign Exchange Rate
163	UMCSENTx	1	1	University of Michigan: Consumer Sentiment (Index 1st Quarter 1966=100)
164	B020RE1Q156NBEA	0	2	Shares of gross domestic product: Exports of goods and services (Percent)
165	B021RE1Q156NBEA	0	2	Shares of gross domestic product: Imports of goods and services (Percent)
166	IPMANSICS	0	5	Industrial Production: Manufacturing (SIC) (Index 2012=100)
167	IPB51222S	0	5	Industrial Production: Residential Utilities (Index 2012=100)
168	IPFUELS	0	5	Industrial Production: Fuels (Index 2012=100)
169	NAPMPI	0	1	ISM Manufacturing: Production Index
170	UEMPMEAN	1	2	Duration of Unemployment (Weeks)
171	CES0600000007	1	2	Average Weekly Hours of Production and Nonsupervisory Employees: Goods-Producing
172	NAPMEI	1	1	ISM Manufacturing: Employment Index
173	NAPM	1	1	ISM Manufacturing: PMI Composite Index
174	NAPMNOI	1	1	ISM Manufacturing: New Orders Index
175	NAPMII	1	1	ISM Manufacturing: Inventories Index
176	TOTRESNS	0	6	Total Reserves of Depository Institutions (Billions of Dollars)
177	GS5	0	2	5-Year Treasury Constant Maturity Rate
178	TB3SMFFM	1	1	3-Month Treasury Constant Maturity Minus Federal Funds Rate
179	T5YFFM	1	1	5-Year Treasury Constant Maturity Minus Federal Funds Rate
180	AAAFFM	1	1	Moodys Seasoned Aaa Corporate Bond Minus Federal Funds Rate
181	PPICRM	1	6	Producer Price Index: Crude Materials for Further Processing (Index 1982=100)
182	PPICMM	0	6	Producer Price Index: Commodities: Metals and metal products: Primary nonferrous metals (Index 1982=100)
183	CPIAPPSL	0	6	Producer Price Index: Commodities: Metals and metal products: Primary nonferrous metals (Index 1982=100)
184	CPITRNSL	1	6	Consumer Price Index for All Urban Consumers: Transportation (Index 1982-84=100)
185	CPIMEDSL	1	6	Consumer Price Index for All Urban Consumers: Medical Care (Index 1982-84=100)
186	CUSR0000SAC	1	6	Consumer Price Index for All Urban Consumers: Commodities (Index 1982-84=100)
187	CUUR0000SAD	1	6	Consumer Price Index for All Urban Consumers: Durables (Index 1982-84=100)
188	CUSR0000SAS	1	6	Consumer Price Index for All Urban Consumers: Services (Index 1982-84=100)
189	CPIULFSL	0	6	Consumer Price Index for All Urban Consumers: All Items Less Food (Index 1982-84=100)
190	CUUR0000SA0L2	0	6	Consumer Price Index for All Urban Consumers: All items less shelter (Index 1982-84=100)
191	CUSR0000SA0L5	0	6	Consumer Price Index for All Urban Consumers: All items less medical care (Index 1982-84=100)
192	CES0600000008	0	6	Average Hourly Earnings of Production and Nonsupervisory Employees: Goods-Producing (Dollars per Hour)
193	DTCOLNVHFNM	0	6	Consumer Motor Vehicle Loans Outstanding Owned by Finance Companies (Millions of Dollars)
194	DTCTHFNM	0	6	Total Consumer Loans and Leases Outstanding Owned and Securitized by Finance Companies (Millions of Dollars)
195	INVEST	1	6	Securities in Bank Credit at All Commercial Banks (Billions of Dollars)
196	HWIURATIO	1	2	Ratio of Help Wanted/No. Unemployed
197	CLAIMSx	1	5	Initial Claims
198	BUSINVx	1	5	Total Business Inventories (Millions of Dollars)
199	ISRATIOx	1	2	Total Business: Inventories to Sales Ratio
200	CONSPI	0	2	Nonrevolving consumer credit to Personal Income
201	CP3M	0	2	3-Month AA Financial Commercial Paper Rate
202	COMPAPFF	0	1	3-Month Commercial Paper Minus Federal Funds Rate
203	PERMITNE	0	5	New Private Housing Units Authorized by Building Permits in the Northeast Census Region (Thousands, SAAR)
204	PERMITMW	0	5	New Private Housing Units Authorized by Building Permits in the Midwest Census Region (Thousands, SAAR)
205	PERMITS	0	5	New Private Housing Units Authorized by Building Permits in the South Census Region (Thousands, SAAR)
206	PERMITW	0	5	New Private Housing Units Authorized by Building Permits in the West Census Region (Thousands, SAAR)
207	NIKKEI225	0	5	Nikkei Stock Average
208	TLBSNNCBx	0	5	Real Nonfinancial Corporate Business Sector Liabilities (Billions of 2009 Dollars), Deflated by IPDBS
209	TLBSNNCBBDIx	0	1	Nonfinancial Corporate Business Sector Liabilities to Disposable Business Income (Percent)
210	TTAABSNNCBx	0	5	Real Nonfinancial Corporate Business Sector Assets (Billions of 2009 Dollars), Deflated by IPDBS
211	TNWMVBSNNCBx	0	5	Real Nonfinancial Corporate Business Sector Net Worth (Billions of 2009 Dollars), Deflated by IPDBS

Table A1 (continued)

212	TNWMVBSNNCBBDIx	0	2	Nonfinancial Corporate Business Sector Net Worth to Disposable Business Income (Percent)
213	NNBTILQ027Sx	0	5	Real Nonfinancial Noncorporate Business Sector Liabilities (Billions of 2009 Dollars), Deflated by IPDBS
214	NNBTILQ027SBDIx	0	1	Nonfinancial Noncorporate Business Sector Liabilities to Disposable Business Income (Percent)
215	NNBTASQ027Sx	0	5	Real Nonfinancial Noncorporate Business Sector Assets (Billions of 2009 Dollars), Deflated by IPDBS
216	TNWBSNNBx	0	5	Real Nonfinancial Noncorporate Business Sector Net Worth (Billions of 2009 Dollars), Deflated by IPDBS
217	TNWBSNNBBDIx	0	2	Nonfinancial Noncorporate Business Sector Net Worth to Disposable Business Income (Percent)
218	CNCFx	0	5	Real Disposable Business Income, Billions of 2009 Dollars
219	S&P 500	1	5	S&Ps Common Stock Price Index: Composite
220	S&P: indust	0	5	S&Ps Common Stock Price Index: Industrials
221	S&P div yield	0	2	S&Ps Composite Common Stock: Dividend Yield
222	S&P PE ratio	0	5	S&Ps Composite Common Stock: Price-Earnings Ratio

B Technical Appendix

B.1 Generalized Approximate Message Passing for Bayesian regression

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 $\beta \sim N_p\left(0, \underline{V}\right)$. Initialize $\mu_\beta^{(1)} = 0$, $\tau_\beta^{(1)} = \underline{V}$, set $s^{(0)} = 0$ and define $S = X \odot X$, where $\odot$ denotes the Hadamard (element-wise) product (also, below, the symbol $\oslash$ is used to denote element-wise division of matrices/vectors of the same dimension). The Generalized Approximate Message Passing (GAMP) algorithm iterates through the following steps (see also Rangan, Schniter, Riegler, Fletcher and Cevher, 2016)

Generalized Approximate Message Passing Algorithm
for r=1:R
$\tau_c^{(r)} = S\tau_\beta^{(r)}$
$c^{(r)} = X\mu_\beta^{(r)} - s^{(r-1)} \odot \tau_c^{(r)}$
$z^{(r)} = E\left(z\middle\vert\ c^{(r)}, \tau_c^{(r)}\right)$
$\tau_z^{(r)} = var\left(z\middle\vert\ c^{(r)}, \tau_c^{(r)}\right)$
$s^{(r)} = \left(z^{(r)} - c^{(r)}\right) \oslash \tau_c^{(r)}$
$\tau_s^{(r)} = \left(1 - \tau_z^{(r)} \oslash \tau_c^{(r)}\right) \oslash \tau_c^{(r)}$
$\tau_q^{(r)} = 1 \oslash \left(S\tau_s^{(r)}\right)$
$q^{(r)} = \mu_\beta^{(r)} + \tau_q^{(r)} \odot X's^{(r)}$
$\mu_\beta^{(r+1)} = E\left(\beta\middle\vert q^{(r)}, \tau_q^{(r)}\right)$
$\tau_\beta^{(r+1)} = var\left(\beta\middle\vert q^{(r)}, \tau_q^{(r)}\right)$
end

The algorithm above involves only multiplications and additions which require a total of $\mathcal{O}(Tp)$ operations. We have written the algorithm above in a form that requires element-wise operations, thus, taking advantage of modern matrix programming languages. Depending on the dimension of the problem, the algorithm above can be written (with "*for*" loops, which are trivial to parallelize) using multiplications and additions of scalar quantities. The reason

for being able to do so, is the assumption of posterior independence of $p\left(\beta_i|\beta_{(-i)}\right) = p\left(\beta_i\right)$, $\forall i = 1, ..., p$.

The algorithm converges when $\|\mu_\beta^{(r+1)} - \mu_\beta^{(r)}\| \to 0$. In this case the posterior of β is

$$\beta|\mathcal{D} \sim N\left(\mu_\beta^{(r+1)}, \tau_\beta^{(r+1)}\right), \tag{B.1}$$

where $\mathcal{D}$ denotes the conditioning information (data y). Assuming σ^2 is known, derivation of the mean and variance of $z|c^{(r)}, \tau_c^{(r)}$ and $\beta|q^{(r)}, \tau_q^{(r)}$ in the GAMP algorithm is based on the assumption of posterior independence. In particular, conditional on σ^2 (and the prior hyperparameters $\underline{\beta}, \underline{V}_\beta$), GAMP assumes posterior independence among the latent parameters β_i, $i = 1, ..., p$ and approximates the true marginal posterior distribution $p\left(\beta_i|\mathcal{D}\right)$ by

$$p\left(\beta_i|\mathcal{D}, q_i, \tau_{q,i}\right) = \frac{p\left(\beta_i\right) N\left(\beta_i|q_i, \tau_{q,i}\right)}{\int_\beta p\left(\beta_i\right) N\left(\beta_i|q_i, \tau_{q,i}\right)} \tag{B.2}$$

where $q_i, \tau_{q,i}$ are the ith elements of the vectors q, τ^q, respectively, which are updated iteratively in the GAMP algorithm above. Given that we have defined $p\left(\beta\right) \sim N_p\left(0, \underline{V}\right)$ above, we can derive $p\left(\beta_i|\mathcal{D}, q_i, \tau_{q,i}\right)$ to be a Normal distribution with location and scale parameters

$$E\left(\beta_i|q_i, \tau_{q,i}\right) = \frac{q_i}{1 + \tau_{q,i}\underline{V}_{ii}}, \tag{B.3}$$

$$var\left(\beta_i|q_i, \tau_{q,i}\right) = \frac{\tau_{q,i}}{1 + \tau_{q,i}\underline{V}_{ii}}. \tag{B.4}$$

Similarly, GAMP assumes posterior independence among $z_t = X_t\beta$, that is, the true marginal posterior $p\left(z_t|\mathcal{D}\right)$ can be approximated by

$$p\left(z_t|\mathcal{D}, c_t, \tau_{c,t}, \sigma^2\right) = \frac{p\left(y_t|z_t, \sigma^2\right) N\left(z_t|c_t, \tau_{c,t}\right)}{\int_z p\left(y_t|z_t, \sigma^2\right) N\left(z_t|c_t, \tau_{c,t}\right)}, \tag{B.5}$$

where $c_t, \tau_{c,t}$ are the tth element of the vectors c, τ_c, respectively, in the algorithm above. The function $p\left(y_t|z_t, \sigma^2\right)$ is simply the likelihood $N\left(y_t|X_t\beta, \sigma^2\right)$. In this case, the marginal

posterior of $p\left(z_t|\mathcal{D}, c_t, \tau_{c,t}, \sigma^2\right)$ is a Normal distribution with location and scale parameters

$$E\left(z_t|c_t, \tau_{c,t}\right) = \frac{\tau_{c,t} y_t \sigma^2 + c_t}{1 + \tau_{c,t}\sigma^2}, \tag{B.6}$$

$$var\left(z_t|c_t, \tau_{c,t}\right) = \frac{\tau_{c,t}}{1 + \tau_{c,t}\sigma^2}. \tag{B.7}$$

B.2 Algorithms for joint estimation of parameters

The previous subsection outlined the GAMP algorithm for estimation of the regression coefficients β, conditional on knowledge of the variance parameter σ^2. In practical situations σ^2 will be a parameter to be estimated. In the case where interest lies on hierarchical Bayes priors (such as the ones proposed in the main body of this paper), then we face additional parameters which (from a Bayesian perspective) are random variables that need to be estimated. Given that the approximations involved in the iterated GAMP algorithm result in vast computational benefits compared to existing MCMC methods for parameter estimation, it would be redundant to combine GAMP with MCMC for joint estimation of parameters in a regression. In this case, and given the iterative nature of GAMP, it is easier to rely on the EM algorithm. Given some additional parameters (i.e. other than β) and hyperparameters, which we can denote by θ, a generic EM-augmented GAMP algorithm is as follows

EM-GAMP algorithm for joint estimation
Initialize $\theta^{(0)}$
for r=1:R
E-step: Update $\beta^{(r)}\|\mathcal{D}, \theta^{(r-1)} \sim N\left(\mu_\beta^{(r+1)}, \tau_\beta^{(r+1)}\right)$ using the GAMP algorithm
M-step: Update $\theta^r\|\mathcal{D}, \beta^{(r)}$ by maximizing the associated Q-function
end

While the E-step of the algorithm above is provided by the GAMP iterations, finding the Q-function and obtaining its maximum in the M-step, sounds quite demanding. However, for many parameters and hyperparameters of interest (such as σ^2) derivation of the M-step of the EM algorithm can heavily resemble existing derivations for MCMC algorithms (that is, conditional posterior means used in popular Gibbs sampler algorithms). I make this point

clear with the following two examples:

1. **Algorithm for Normal-Gamma prior (Sparse Bayesian Learning, SBL)**: I assume the following prior structure

$$p\left(\beta_i|\alpha_i\right) \sim N\left(0,\alpha_i\right), \tag{B.8}$$

$$p\left(\alpha_i^{-1}\right) \sim Gamma\left(\underline{a},\underline{b}\right), \tag{B.9}$$

$$p\left(\sigma^{-2}\right) \sim Gamma\left(\underline{c}_1,\underline{c}_2\right), \tag{B.10}$$

such that $\beta \sim N_p\left(0,\underline{V}\right)$ with $\underline{V} = diag(\alpha_1,\alpha_2,...,\alpha_p)$. Optimizing with respect to α means finding the maximum of the following Q-function

$$\alpha^{(r+1)} = \arg\max_{\alpha} Q\left(\alpha|\alpha^{(r)}\right) \equiv E_{\alpha^{(r)}}\left[\log p\left(\alpha|y,\beta^{(r)}\right)\right]. \tag{B.11}$$

Taking the derivative of the Q-function w.r.t. α_i and setting it to zero gives the usual formula from MCMC-based updates of α

$$\alpha_i^{(r+1)} = \frac{2\underline{a}-1}{2\underline{b}+\left(\mu_{i,\beta}^{(r)}\right)^2}. \tag{B.12}$$

where recall that $\mu_{i,\beta}^{(r)} = E\left(\beta_i|q^{(r)},\tau_q^{(r)}\right)$ is output of the algorithm outlined in the previous subsection. Following a similar procedure for σ^{-2} gives

$$\left(\sigma^{-2}\right)^{(r+1)} = \arg\max_{\sigma^{-2}} E_{\beta^{(r)}}\left[\log p\left(y|X,\beta^{(r)},\left(\sigma^{-2}\right)^{(r)}\right)p\left(\sigma^2\right)\right] \tag{B.13}$$

$$= \frac{T+2\underline{c}_1-2}{2\underline{c}_2+\sum_{t=1}^{T}\left(y_t-X_t\mu_{\beta}^{(r)}\right)^2}. \tag{B.14}$$

In this case, the M-step for this case consists of the updating equations (B.12) - (B.14), and then conditional on these steps one can update $\beta^{(r+1)}$, conditional on $\alpha^{(r+1)}$, $\left(\sigma^{-2}\right)^{(r+1)}$, using the GAMP iterations of the previous section.

2. **Algorithm for spike and slab prior (SNS)**: We assume the following prior structure

$$p\left(\beta_i|\pi_0,\alpha\right) \sim (1-\pi_0)\delta_0 + \pi_0 N\left(0,\alpha\right), \tag{B.15}$$

$$p\left(\sigma^{-2}\right) \sim Gamma\left(\underline{c}_1,\underline{c}_2\right). \tag{B.16}$$

Derivation of the EM algorithm can be found in Vila and Schniter (2011,2013). The posterior of β_i is of the form

$$p\left(\beta_i|y,\pi_0\right) = (1-\pi_i)\delta_0 + \pi_i N\left(\mu_{\beta,i}^{(r)},\tau_{\beta,i}^{(r)}\right), \tag{B.17}$$

where we define $\tau_{\beta,i}^{(r)} = \left(\alpha^{-1} + \left(\tau_{q,i}^{(r)}\right)^{-1}\right)^{-1}$, $\mu_{\beta,i}^{(r)} = \tau_{\beta,i}^{(r)} \times \left(q_i^{(r)}/\tau_{q,i}^{(r)}\right)$ and

$$\pi_i^{(r)} = \frac{1}{1+\frac{\left(1-\pi_0^{(r-1)}\right)N(0|q_i^{(r)},\tau_{q,i}^{(r)})}{\pi_0 N(0|-q_i^{(r)},\alpha^{(r-1)}+\tau_{q,i}^{(r)})}}, \tag{B.18}$$

is the posterior probability of inclusion of predictor i in the regression model, $\pi_i \in [0,1]$. For the update step of the variance parameter σ^2 one can follow the derivations of Vila and Schniter in order to obtain an EM update. However, based on extensive experiments I have found that an estimate of the variance based on an approximation to the posterior mode works best

$$\left(\sigma^2\right)^{(r)} = \left(2\underline{c}_2 + \sum_{i=1}^{T}(y_i - z_i^{(r)})^2\right) / \left(2\underline{c}_1 + T\right). \tag{B.19}$$

Finally, remaining parameters can also be updated online (in each iteration) based on EM algorithm derivations:

$$\pi_0^{(r)} = \frac{1}{p}\sum_{i=1}^{p}\pi_i^{(r)}, \tag{B.20}$$

$$\alpha^{(r)} = \frac{1}{\pi_0^{(r)}} \times \frac{1}{p}\sum_{i=1}^{p}\pi_i^{(r)}\left(\mu_{\beta,i}^{(r)}\right)^2 + \tau_{\beta,i}^{(r)}. \tag{B.21}$$

B.3 Incorporating stochastic volatility in GAMP

The main model specification has followed closely research on forecasting with many predictors; see Stock and Watson (2012) and references therein. However, it is well documented that macroeconomic volatility is such an important modelling assumption, that should hardly be ignored from any forecasting or structural model; see for example Clark and Ravazzollo (2015) and the arguments in Clark (2011). In many Bayesian macroeconomic applications involving MCMC-based estimation it is typical to consider a stochastic volatility specification for log-variances, since the additional computation involved in doing so is trivial (just a run of a univariate Kalman filter). However, since this paper is all about how to implement fast computation of shrinkage estimators in high-dimensions, it would be superfluous to combine GAMP with an MCMC or sequential Monte Carlo update for stochastic volatility. Additionally, it is not straightforward to derive a GAMP-based approximations to the estimation of the stochastic volatility volatility specification [10].

Therefore, here I follow a simpler strategy in order to allow for changing variances. Estimation is based on Exponential Weighted Moving Average (EWMA) filters. One approach would be to apply the typical EWMA formula

$$\sigma_t^2 = \kappa\sigma_{t-1}^2 + (1-\kappa)\varepsilon_t^2, \tag{B.22}$$

where $0 < \kappa < 1$ is a decay factor controlling the (exponentially decaying) effective window of observations used in estimation of σ_t^2. It is well-known that such a model approximates an integrated GARCH(1,1) specification, and has been very successful in finance for many decades; see also Koop and Korobilis (2012) for a macroeconomic application. Algorithmically, because of the iterative nature of GAMP, one could run the first iteration conditional on a fixed volatility estimate (e.g. fix $\sigma_t^2 = 1 \ \forall t \in [1, T]$), but from the second iteration onward update σ_t^2 with the residuals from the previous iteration, $\left(\varepsilon_t^{(r-1)}\right)^2$, plugged in the formula

[10] As an example of how complicated Belief Propagation and GAMP would become in a dynamic setting, see for instance Ziniel and Schniter (2013) and Ziniel, Potter and Schniter (2010). The algorithms proposed in these papers involve several approximations and many tedious steps that lack the simplicity and numerical stability of the GAMP algorithms proposed so far.

above.

Alternatively, one can follow West and Harrison (1997) and specify a Beta evolution process of the form

$$\sigma_t^{-2} = \gamma_t \sigma_{t-1}^{-2}/\delta, \quad \gamma_t \sim Beta\left(\kappa n_{t-1}/2, (1-\kappa)\, n_{t-1}/2\right), \tag{B.23}$$

where n_t are the time t degrees of freedom of the process, and κ is also a decay factor. Following results in West and Harrison (1997), it is straightforward to show that given the random initial condition $\sigma_0^{-2} \sim Gamma\left(n_0/2, d_0/2\right)$, then σ_t^{-2} is distirbuted Gamma (or equivalently, σ_0^2 follows the inverse Gamma). This specification is essentially quite similar to the EWMA specification above, but the main difference is that it provides a full posterior for σ_t^2 instead of a point estimate.

The results in Table B1 illustrate the benefits of enhancing GAMP with a stochastically evolving variance. The same regressions are run as in the main text, but now the variance evolves according to the EWMA specification in eq. (B.22). In order to avoid selecting κ subjectively, or optimize it using possibly unstable numerical methods, I follow a simple scheme where I run several instances of GAMP in parallel with $\kappa \in [0.94, 0.98]$ with a step of 0.001 providing a grid of 41 values of this scalar parameter[11]. I then do a simple model averaging with equal weights in order to forecast with the weighted average of all 41 models, despite the fact that one can also consider more sophisticated schemes (e.g. Koop and Korobilis, 2012, use predictive likelihoods to calculate time-varying weights). Given the fast nature of GAMP and the simple recursive updates of the EWMA filter, which only involves scalar multiplications and additions, there is no noticeable increase in computing times from this extension. The first two columns of Table B1 show the same GAMP models with added EWMA volatility, denoted SBL-EWMA and SNS-EWMA, respectively. We can immediately observe an obvious shift to the left for the distributions of relative RMSEs of both the SBL-EWMA and SNS-EWMA cases, compared to the models with constant variance.

[11] See details in Koop and Korobilis (2012) why these values are plausible in this settings

Table B1: Distributions of Relative Root Mean Squared Errors (RMSE), Relative to the AR(1) Forecast, by Forecasting Method, all 222 series

		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
0.075	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

Notes: Entries are percentiles of distributions of relative RMSFEs over the 222 variables being forecasted, by series, at the 1-, 2-, 4- and 8-quarter ahead forecast horizon. RMSFEs are relative to the AR(1) forecast RMSFE, and are computed using an expanding window of in-sample observations (recursive). All forecasts are direct.

B.4 Details of competing MCMC-based algorithms

In the main body of the paper, both in the Monte Carlo study and the empirical exercise, the Bayesian lasso and the stochastic search variable selection (SSVS) have been used as benchmarks. This subsection describes these two priors and the choice of relevant hyperparameters

B.4.1 The Bayesian lasso

The full hierarchical representation of the lasso prior is

$$\begin{aligned} p\left(\beta|\sigma^2,\tau_1^2,...,\tau_p^2\right) &\sim N\left(0,\sigma^2 V\right), && \text{(B.24a)}\\ p\left(\tau_j^2\right) &\sim Exponential\left(\frac{\lambda^2}{2}\right), \text{ for } j=1,...,p, && \text{(B.24b)}\\ p\left(\lambda^2\right) &\sim Gamma\left(r,\delta\right), && \text{(B.24c)}\\ p\left(\sigma^2\right) &\propto 1/\sigma^2, && \text{(B.24d)} \end{aligned}$$

where $V = diag\left\{\tau_1^2,...,\tau_p^2\right\}$. I follow the recommendations in Park and Casella (2008) and I choose $r = 1$ and $\delta = 3$.

Given these priors, the posterior can be obtained by sampling recursively from the conditional posteriors

$$\begin{aligned} \beta|\sigma^2,\left\{\tau_j^2\right\}_{j=1}^p,y &\sim N\left(\left(x'x+V^{-1}\right)^{-1}x'y,\sigma^2\left(x'x+V^{-1}\right)^{-1}\right), && \text{(B.25a)}\\ \frac{1}{\tau_j^2}|\beta,\sigma^2,y &\sim IG\left(\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}},\lambda^2\right), \text{ for } j=1,...,p, && \text{(B.25b)}\\ \lambda^2|\beta,\sigma^2,\left\{\tau_j^2\right\}_{j=1}^p,y &\sim Gamma\left(p+r,\frac{1}{2}\sum\nolimits_{j=1}^p\tau_j^2+\delta\right), && \text{(B.25c)}\\ \sigma^2|\beta,\left\{\tau_j^2\right\}_{j=1}^p,y &\sim iGamma\left(\frac{T-1}{2}+\frac{p}{2},\frac{1}{2}\Psi+\frac{1}{2}\beta'V^{-1}\beta\right), && \text{(B.25d)} \end{aligned}$$

where IG denotes the Inverse-Gaussian distribution.

B.4.2 Stochastic search variable selection

The full hierarchical representation of the SSVS prior is

$$p\left(\beta_i|\gamma_i\right) \sim (1-\gamma_i)N\left(0,\tau_0^2\right)+\gamma_i N\left(0,\tau_1^2\right), \tag{B.26a}$$

$$p\left(\gamma_i|\pi\right) \sim Bernoulli(\pi_i), \tag{B.26b}$$

where I set $\pi = 0.5$, and $\tau_0 = 0.001$ and $\tau_1 = 4$.

Given these priors, the posterior can be obtained by sampling recursively from the conditional posteriors

$$\beta|\sigma^2,\gamma,y \sim N\left(\left(x'x/\sigma^2+V^{-1}\right)^{-1}x'y/\sigma^2,\left(x'x/\sigma^2+V^{-1}\right)^{-1}\right), \tag{B.27a}$$

$$\gamma_i|\beta_i,\pi,y \sim Bernoulli\left(\frac{N(0|\beta_i,\tau_1^2)\pi}{N(0|\beta_i,\tau_0^2)(1-\pi)+N(0|\beta_i,\tau_1^2)\pi}\right), \tag{B.27b}$$

$$\sigma^2|\beta,y \sim iGamma\left(\frac{T}{2},\frac{1}{2}\sum_{t=1}^{T}(y_t-x_t\beta)^2\right), \tag{B.27c}$$

where V is a diagonal matrix with i-th element τ_0^2 if $\gamma_i = 0$ or τ_1^2 if $\gamma_i = 1$.

C Additional Results

C.1 Results using only a subset of "important" variables

Table C1: Distributions of Relative Root Mean Squared Errors (RMSE), Relative to the AR(1) Forecast, by Forecasting Method, 14 selected series

				Horizon $h = 1$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.751	0.847	0.779	0.724	0.729	0.830	0.746	0.957
0.25	0.888	0.909	0.901	0.877	0.921	0.909	0.922	1.073
0.5	0.963	0.993	1.002	0.968	0.970	0.980	0.966	1.166
0.075	1.007	1.001	1.032	1.000	1.013	1.026	1.036	1.338
0.95	1.092	1.051	1.145	1.049	1.067	1.155	1.143	1.624
				Horizon $h = 2$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.784	0.769	0.832	0.766	0.815	0.857	0.767	0.964
0.25	0.838	0.887	0.860	0.839	0.864	0.917	0.829	1.064
0.5	0.976	0.995	1.034	0.970	0.988	1.007	1.031	1.198
0.75	1.062	1.007	1.052	1.019	1.065	1.055	1.117	1.369
0.95	1.112	1.122	1.148	1.068	1.114	1.307	1.275	2.013
				Horizon $h = 4$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.806	0.769	0.830	0.766	0.785	0.865	0.782	0.979
0.25	0.875	0.888	0.880	0.870	0.857	0.936	0.827	1.053
0.5	0.937	0.971	0.931	0.956	0.948	0.974	0.965	1.149
0.75	1.069	1.000	1.105	1.021	1.109	1.097	1.075	1.492
0.95	1.171	1.017	1.187	1.114	1.230	1.165	1.248	1.680
				Horizon $h = 8$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.778	0.769	0.763	0.752	0.761	0.802	0.732	0.967
0.25	0.949	0.951	0.944	0.949	0.960	1.010	0.898	1.109
0.5	1.035	1.000	1.061	1.009	1.032	1.063	0.950	1.235
0.75	1.067	1.023	1.108	1.029	1.057	1.118	1.096	1.497
0.95	1.151	1.097	1.331	1.136	1.321	1.382	1.286	1.930

Notes: Entries are percentiles of distributions of relative RMSFEs over the 14 variables being forecasted, by series, at the 1-, 2-, 4- and 8-quarter ahead forecast horizon. RMSFEs are relative to the AR(1) forecast RMSFE, and are computed using an expanding window of in-sample observations (recursive). All forecasts are direct.

Table C2: Multivariate weighted mean squared forecast error, for each estimator and forecast horizon, 14 selected series

	$h=1$	$h=2$	$h=4$	$h=8$
SBL	0.940	0.976	1.006	1.023
SNS	0.959	0.932	0.909	0.973
LASSO	0.987	1.038	1.035	1.110
SSVS	0.924	0.939	0.945	0.981
BMA	0.960	0.996	1.019	1.079
BAG	0.997	1.057	1.057	1.194
DFM5	0.964	1.027	1.011	0.999
OLS	1.555	1.624	1.594	1.897

Notes: Entries in this table report the multivariate weighted mean squared forecast error of each method relative to the AR(1); see Christoffersen and Diebold (1998) for more details.

Table C3: Two Measures of Similarity of Forecast Performance, $h=1$: Correlation (lower left) and Mean Absolute Difference of Forecasts (upper right), 14 selected series

	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
SBL		0.057	0.035	0.027	0.022	0.040	0.034	0.269
SNS	0.50		0.057	0.043	0.059	0.055	0.072	0.253
LASSO	0.90	0.65		0.049	0.040	0.028	0.043	0.241
SSVS	0.88	0.67	0.79		0.025	0.060	0.043	0.284
BMA	0.94	0.57	0.89	0.94		0.045	0.034	0.264
BAG	0.89	0.65	0.96	0.74	0.85		0.049	0.233
DFM5	0.69	0.52	0.88	0.49	0.69	0.88		0.250
OLS	0.86	0.36	0.81	0.71	0.76	0.81	0.67	

Notes: Entries below the diagonal are the correlation between the RMSFEs for the row/column forecasting methods, computed over the 14 series being forecasted. Entries above the diagonal are the mean absolute difference between the row/column method RMSFEs, averaged across series.

Table C4: Hit-rates of the eight estimators, 14 selected series

				Horizon $h = 1$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	0.00	7.14	14.29	42.86	7.14	7.14	21.43	0.00
% of lowest RMSFE	21.43	21.43	7.14	14.29	14.29	21.43	0.00	0.00
				Horizon $h = 2$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	28.57	42.86	7.14	14.29	0.00	0.00	7.14	0.00
% of lowest RMSFE	14.29	35.71	7.14	7.14	7.14	7.14	21.43	0.00
				Horizon $h = 4$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	7.14	35.71	7.14	14.29	7.14	0.00	28.57	0.00
% of lowest RMSFE	7.14	28.57	14.29	21.43	7.14	0.00	21.43	0.00
				Horizon $h = 8$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	7.14	28.57	7.14	14.29	0.00	0.00	42.86	0.00
% of lowest RMSFE	0.00	28.57	14.29	0.00	0.00	7.14	50.00	0.00

Note: This table shows the proportion of times (over the 14 series being forecasted) that each estimator achieved the lowest value of the MAFE and RMSFE statistics.

C.2 Results using rolling forecasts

Table C5: Distributions of Relative Root Mean Squared Errors (RMSE), Relative to the AR(1) Forecast, by Forecasting Method, rolling forecasts, 222 selected series

				Horizon $h = 1$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.800	0.801	0.858	0.794	0.808	0.793	0.823	0.923
0.25	0.918	0.952	0.970	0.943	0.948	0.953	0.935	1.223
0.5	0.994	0.999	1.019	0.990	0.996	1.025	0.999	1.409
0.075	1.018	1.003	1.058	1.010	1.022	1.075	1.045	1.553
0.95	1.049	1.022	1.160	1.044	1.084	1.187	1.115	1.822
				Horizon $h = 2$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.785	0.794	0.871	0.801	0.804	0.768	0.835	0.895
0.25	0.904	0.934	0.962	0.916	0.926	0.925	0.918	1.202
0.5	0.985	0.997	1.019	0.991	0.998	1.007	0.995	1.395
0.75	1.018	1.005	1.077	1.012	1.037	1.086	1.049	1.570
0.95	1.073	1.034	1.166	1.063	1.108	1.156	1.131	1.872
				Horizon $h = 4$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.744	0.773	0.841	0.797	0.818	0.767	0.834	0.871
0.25	0.892	0.912	0.940	0.902	0.912	0.914	0.906	1.181
0.5	0.980	0.987	1.019	0.982	0.991	1.015	0.994	1.399
0.75	1.022	1.004	1.091	1.012	1.043	1.104	1.061	1.611
0.95	1.077	1.048	1.194	1.079	1.123	1.203	1.163	1.998
				Horizon $h = 8$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
0.05	0.621	0.601	0.789	0.732	0.749	0.774	0.790	0.966
0.25	0.911	0.927	0.937	0.909	0.910	0.942	0.910	1.185
0.5	0.986	0.989	1.050	0.988	1.008	1.055	0.997	1.396
0.75	1.048	1.014	1.117	1.038	1.075	1.145	1.088	1.658
0.95	1.130	1.079	1.236	1.133	1.179	1.274	1.234	2.123

Notes: Entries are percentiles of distributions of relative RMSFEs over the 14 variables being forecasted, by series, at the 1-, 2-, 4- and 8-quarter ahead forecast horizon. RMSFEs are relative to the AR(1) forecast RMSFE, and are computed using a fixed window of 100 in-sample observations (rolling). All forecasts are direct.

Table C6: Multivariate weighted mean squared forecast error, for each estimator and forecast horizon, rolling forecasts, 222 selected series

	$h=1$	$h=2$	$h=4$	$h=8$
SBL	0.920	0.914	0.865	0.876
SNS	0.928	0.962	1.072	0.856
LASSO	0.993	1.039	1.192	1.174
SSVS	0.932	0.987	1.134	0.927
BMA	0.989	1.053	1.218	1.150
BAG	0.931	0.910	0.868	0.902
DFM5	0.932	0.892	0.911	0.837
OLS	1.074	1.067	1.248	1.511

Notes: Entries in this table report the multivariate weighted mean squared forecast error of each method relative to the AR(1); see Christoffersen and Diebold (1998) for more details.

Table C7: Two Measures of Similarity of Forecast Performance, rolling forecasts, $h = 1$: Correlation (lower left) and Mean Absolute Difference of Forecasts (upper right), 222 selected series

	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
SBL		0.026	0.054	0.024	0.027	0.092	0.046	0.418
SNS	0.93		0.050	0.028	0.034	0.091	0.054	0.412
LASSO	0.87	0.85		0.056	0.047	0.073	0.059	0.376
SSVS	0.87	0.87	0.81		0.017	0.096	0.042	0.419
BMA	0.86	0.83	0.84	0.97		0.086	0.038	0.408
BAG	0.61	0.51	0.66	0.52	0.59		0.088	0.377
DFM5	0.16	0.03	0.26	0.12	0.20	0.67		0.399
OLS	0.38	0.28	0.40	0.48	0.52	0.46	0.49	

Notes: Entries below the diagonal are the correlation between the RMSFEs for the row/column forecasting methods, computed over the 222 series being forecasted. Entries above the diagonal are the mean absolute difference between the row/column method RMSFEs, averaged across series.

Table C8: Hit-rates of the eight estimators, rolling forecasts, 222 selected series

				Horizon $h = 1$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	10.81	24.32	3.15	19.82	8.11	15.32	15.77	2.70
% of lowest RMSFE	10.36	23.42	1.80	16.22	9.01	15.32	20.72	3.15
				Horizon $h = 2$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	14.41	29.28	2.70	11.71	4.50	16.67	18.02	2.70
% of lowest RMSFE	14.41	28.83	1.80	12.16	4.05	15.77	19.82	3.15
				Horizon $h = 4$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	13.51	29.73	2.70	13.51	6.31	13.51	19.37	1.35
% of lowest RMSFE	14.41	28.38	3.60	11.26	4.96	8.11	26.58	2.70
				Horizon $h = 8$				
	SBL	**SNS**	**LASSO**	**SSVS**	**BMA**	**BAG**	**DFM5**	**OLS**
% of lowest MAFE	14.87	28.83	2.25	9.91	2.25	9.46	31.98	0.45
% of lowest RMSFE	12.16	29.28	3.60	8.11	2.70	9.01	34.23	0.90

Note: This table shows the proportion of times (over the 222 series being forecasted) that each estimator achieved the lowest value of the MAFE and RMSFE statistics.