JOURNAL OF BUSINESS & ECONOMIC STATISTICS
2019, VOL. 00, NO. 0, 1–12
https://doi.org/10.1080/07350015.2019.1677472

High-Dimensional Macroeconomic Forecasting Using Message Passing Algorithms

Dimitris Korobilis

Adam Smith Business School, University of Glasgow, Glasgow, UK

ABSTRACT

This article proposes two distinct contributions to econometric analysis of large information sets and structural instabilities. First, it treats a regression model with time-varying coefficients, stochastic volatility, and exogenous predictors, as an equivalent high-dimensional static regression problem with thousands of covariates. Inference in this specification proceeds using Bayesian hierarchical priors that shrink the high-dimensional vector of coefficients either toward zero or time-invariance. Second, it introduces the frameworks of factor graphs and message passing as a means of designing efficient Bayesian estimation algorithms. In particular, a generalized approximate message passing algorithm is derived that has low algorithmic complexity and is trivially parallelizable. The result is a comprehensive methodology that can be used to estimate time-varying parameter regressions with arbitrarily large number of exogenous predictors. In a forecasting exercise for U.S. price inflation this methodology is shown to work very well. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received May 2017
Accepted September 2019

KEYWORDS
Bayesian shrinkage; Belief propagation; Factor graph; High-dimensional inference; Time-varying parameter model

1. Introduction

As a response to the increasing linkages between the macroeconomy and the financial sector, as well as the expanding interconnectedness of the global economy, empirical macroeconomic models have increased both in complexity and size. For that reason, estimation of modern models that inform macroeconomic decisions—such as linear and nonlinear versions of dynamic stochastic general equilibrium (DSGE) and vector autoregressive (VAR) models—many times relies on Bayesian inference via powerful Markov chain Monte Carlo (MCMC) methods.[1] However, existing posterior simulation algorithms cannot scale up to very high-dimensions due to the computational inefficiency and the larger numerical error associated with repeated sampling via Monte Carlo; see Angelino, Johnson, and Adams (2016) for a thorough review of such computational issues from a machine learning and high-dimensional data perspective. In that respect, while Bayesian inference is a natural probabilistic framework for learning about parameters by using all information in the data likelihood and prior, computational restrictions might make it less suitable for supporting real-time decision-making in very high dimensions.

This article introduces to the econometric literature the framework of factor graphs (Kschischang, Frey, and Loeliger 2001) for the purpose of designing computationally efficient, and easy to maintain, Bayesian estimation algorithms. The focus is not only on "faster" posterior inference broadly interpreted, but on designing algorithms that have such low complexity that are future-proof and can be used in high-dimensional econometric problems with possibly thousands or millions of coefficients. While a graph, in general, is a structure that allows the representation of objects that are related in some sense,[2] a factor graph representation of a high-dimensional vector of model parameters, in particular, depicts how each of its scalar elements is connected with each other based on the functional form of their joint posterior distribution. As a result, the factor graph representation provides a visual tool for the decomposition of a high-dimensional joint posterior distribution into smaller, tractable parts. By doing so, factor graphs can be used to design parallel versions of MCMC algorithms, as well as efficient iterative algorithms called *message passing algorithms*—the latter being the concept of interest in this article.[3]

Having the factor graph as the starting point, interest lies in an estimation strategy called the sum-product algorithm which is not well known in mainstream statistics, despite the fact that it is computationally powerful (Wand 2017, pp. 137–138). The sum-product algorithm is a general rule in factor graphs that allows to iteratively approximate marginal (posterior) distributions. When applied to a parametric problem with arbitrary likelihood and prior functions, the so-called generalized approximate message passing (GAMP) algorithm introduces further Gaussian and quadratic approximations to the possibly complicated expressions derived by the sum-product iterative algorithm. Proposed by Rangan (2011), GAMP is an extension of the popular approximate message passing (AMP) algorithm of Donoho, Maleki, and Montanari (2009). The GAMP algorithm

CONTACT Dimitris Korobilis *dikorobilis@googlemail.com* Adam Smith Business School, University of Glasgow, Glasgow G12 8QQ, UK.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/UBES*.

[1] See Herbst and Schorfheide (2015) and Koop and Korobilis (2010) for detailed discussion of Bayesian computation in DSGE and VAR models, respectively.

[2] The most popular use of graphs in economics is to represent networks of agents, banks, social networks, etc.; see Jackson (2008).

[3] Message passing algorithms are dynamic programming methods designed for efficiently performing large computations by distributing calculations among a number of simpler processors. Readers working with high-performance clusters (HPC) might be familiar with the related concept of message passing interface (MPI) which is a standardized means for exchanging data/commands between multiple processors in a computer cluster.

has desirable properties, namely, high-dimensional scalability, parallelizability, and effortless maintenance. Therefore, the first task of this article is to analyze the concept of message passing algorithms in general; simplify the jargon stemming from signal processing, computing science, and similar literatures that have introduced such algorithms; and show how GAMP, in particular, can lead to efficient posterior inference in very high-dimensions.

At the same time, a second important task is to provide compelling evidence that the proposed algorithm is relevant for modeling macroeconomic variables. For that reason, I utilize a regression model setting with time-varying coefficients, stochastic volatility, and exogenous predictors. Regression models featuring time-varying parameters (TVPs) have been popular in economics at least since the seminal work of Cooley and Prescott (1976). More recently, there has been a systematic effort to introduce efficient MCMC algorithms for flexible estimation and shrinkage in Bayesian TVP models (see Koop and Potter 2007; Stock and Watson 2007; Giordani and Kohn 2008; Chan et al. 2012; Groen, Paap, and Ravazzollo 2013; Nakajima and West 2013; Belmonte, Koop, and Korobilis 2014; Kalli and Griffin 2014; Kowal, Matteson, and Ruppert 2019; Ročková and McAlinn 2018, among others). These are examples of carefully designed MCMC algorithms that result in flexible joint modeling of structural instabilities and parameter shrinkage, but that may not be scalable to very high dimensions due to their reliance on repeated sampling via Monte Carlo.

As a consequence, a novel empirical contribution introduced in this article is to estimate a TVP regression model by using an observationally equivalent high-dimensional static regression form, and to address computational concerns by using message passing inference. With T observations and p predictors, the TVP model can be written as a static regression with the same T observations but $(T+1)p$ covariates—where the product $(T+1)p$ can easily be in the order of tens of thousands in standard macroeconomic applications. This static representation of the TVP model is anything but new, however, its estimation in the past has been exclusively tackled by specifying an additional hierarchical random walk (or sometimes stationary autoregressive) model for all TVPs. This hierarchical form allows for inference using state-space methods and at the same time it can be interpreted as an informative shrinkage prior that makes estimation of this high-dimensional problem feasible. Instead I propose to completely drop this "random-walk prior" and the resulting state-space representation, and estimate the TVP model as a high-dimensional static regression with the assistance of a flexible Bayesian hierarchical shrinkage prior inspired by Tipping (2001). That way, by casting the TVP regression model into equivalent static form, standard shrinkage principles can be used to determine by how much coefficients evolve over time, or whether their value is zero and they are completely irrelevant. Most importantly, the use of the low-complexity GAMP algorithm ensures that the static form of the TVP regression with $(T+1)p$ covariates can be estimated quickly. The benefits of this algorithm and modeling strategy are illustrated using a forecasting exercise for monthly U.S. inflation that extends Stock and Watson (1999) to the TVP setting. The static form of the TVP regression estimated with GAMP is contrasted with powerful but slow MCMC algorithms for TVP models, such as Chan et al. (2012) and Kalli and Griffin (2014). The proposed approach, by incorporating a larger number of predictors and by shrinking coefficients flexibly, does perform significantly better compared to competitors in out-of-sample forecasting.

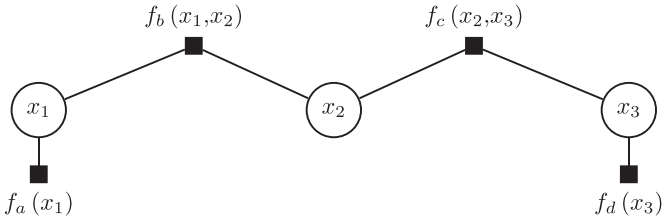

Figure 1. Simple factor graph representation of the decomposition of joint function $p(x_1, x_2, x_3)$.

In the next section, I introduce the general framework of factor graphs on random variables (parameters) and with the help of a toy example I show how this framework allows for efficient calculation of marginal distributions. Next, in Section 3, I introduce the TVP regression setting, rewrite the likelihood in static regression form and specify a shrinkage "sparse Bayesian learning" (SBL) prior. Under the given functional forms for the likelihood and prior, I proceed to derive a GAMP algorithm for this particular problem. In Section 4, the benefits of the proposed high-dimensional modeling approach are evaluated in a forecasting exercise for U.S. price inflation. Section 5 concludes the article.

2. Factor Graphs and the Sum-Product Algorithm

A factor graph represents the way a global function of several variables can be decomposed into a product of simpler functions ("factors"). Consider a generic example with *discrete* random variables $x = (x_1, x_2, x_3)$ and a joint mass function p that we can decompose, say, as

$$p(x_1, x_2, x_3) = f_a(x_1) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3), \quad (1)$$

where f_a, f_b, f_c, f_d are the factors that have known functional forms.[4] This simple example can be depicted using the factor graph of Figure 1, where circles denote the place of random variables in the graph and filled boxes denote the factors/functions.[5]

Consider now calculation of the marginal distribution of x_i. This is a computationally demanding task due to the fact that it involves integration (summation, in the discrete variable case) over all variables other than x_i

$$p(x_i) = \sum_{x \setminus x_i} p(x_1, x_2, x_3), \quad (2)$$

where $x \setminus x_i$ denotes the set x with the element x_i removed. As an example, if the variables in x have two states (e.g., they are binary variables), then the above sum would only require 2^3 operations. However, for high number of states and/or variables computational requirements proliferate substantially. Nevertheless, if

[4] In the next section, the discrete random variables x are replaced by continuous model parameters, and the factors/functions are conditional or marginal probability distributions over these parameters.

[5] In graph theory, symbols like the boxes and the circles in this example are called nodes or vertices. Nodes that depend to each other are connected with a solid line, and each connected pair of nodes is called an "edge."

$p(x_1, x_2, x_3)$ is replaced with the expression in (1) it can be seen that not each variable is coupled to every other one, and this feature can be exploited to simplify the summation. For example, in the case of variable x_1, Figure 1 depicts that it is directly connected to x_2 and the factors $f_a(x_1)$ and $f_d(x_1, x_2)$, but it is only indirectly connected to x_3 and the remaining factors. Put differently, we can simplify (2) via identity (1) as follows

$$p(x_1) = \sum_{x_2} \sum_{x_3} f_a(x_1) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3), \quad (3)$$

$$= f_a(x_1) \sum_{x_2} f_b(x_1, x_2) \sum_{x_3} f_c(x_2, x_3) f_d(x_3). \quad (4)$$

The second line of the equation above implies less algorithmic operations compared to the expression in the first line.

It should be clear at this point that the role of the factor graph representation is to allow to pin down the full path of influence that each variable x_i exerts on other variables. As a consequence, by having this path of influence, only the required factors f_j can be used when calculating marginal distributions, which increases computational efficiency. This is where the concept of *message passing* formalizes such an efficient procedure for computing marginals. Each variable node passes messages to the next variable, where these messages are real-valued functions showing the influence that this variable exerts on all other variables. In the remainder of this section, message passing inference is introduced and the sum-product algorithm is derived, such that simplifications similar to the ones in Equations (3) and (4) are formalized mathematically. Subsequently, in Section 3, the results of this toy example with three discrete random variables (parameters) can be generalized to a high-dimensional regression setting with possibly millions of parameters. More detailed introductions to these concepts can be found in popular machine learning textbooks, such as Barber (2012) and Bishop (2006). A recent introduction of message passing inference in factor graphs from a statistician's perspective is provided in Wand (2017).

Denote with $\mu_{x_i \to f_j}$ the message sent from variable x_i to function f_j, and with $\mu_{f_j \to x_i}$ the message sent from factor node f_j to variable node x_i, where $i = 1, 2, 3$ and $j = a, b, c, d$ in our simple example with three variables and four factors. The message sent from variable x_i to factor node f_j is equal to the product of all messages arriving to node x_i except from the message coming from the target node f_j:

$$\mu_{x_i \to f_j} = \prod_{k \in N(x_i), k \neq j} \mu_{f_k \to x_i}, \quad (5)$$

where $N(x_i)$ is the set of neighboring (factor) nodes to x_i. Similarly, the message sent from factor node f_j to variable node x_i is given by the sum over the product of the factor function f_j itself and all the incoming messages, except the messages from the target variable node x_i:

$$\mu_{f_j \to x_i} = \sum_{x \setminus x_i} f_j(x) \prod_{l \in N(x_i), l \neq i} \mu_{x_l \to f_j}. \quad (6)$$

Due to the form of the equation above, algorithms that are designed to iterate between (5) and (6) are called *sum-product* algorithms; see also respective equations for the regression model in the next section.

In the special case where x_i is an external node (as is the case with x_1 and x_3 in this example) it holds that $\mu_{x_i \to f_j} = 1$. Similarly, if f_j is an external factor node (see $f_a(x_1)$ and $f_d(x_3)$ in Figure 1) it holds that $\mu_{f_j \to x_i} = f_j(x_i)$. Equations (5) and (6) define the iterations of the so-called sum-product algorithm (also called belief propagation; see Pearl 1982), that allows calculation of marginal distributions (also called "beliefs" in computing science and the Bayesian networks literature). Upon convergence, it can be shown[6] that

$$p(x_i) \propto \prod_{m \in N(x_i)} \mu_{f_m \to x_i}, \quad (7)$$

that is, the marginal distribution of variable x_i is simply the product of all messages received only from factor nodes that are connected to x_i.

Consider for example calculation of $p(x_2)$. Starting from the left of the graph, the messages emitted to node x_2 are

$$\mu_{f_a \to x_1} = f_a(x_1), \quad (8)$$

$$\mu_{x_1 \to f_b} = \mu_{f_a \to x_1} = f_a(x_1), \quad (9)$$

$$\mu_{f_b \to x_2} = \sum_{x_1} f_b(x_1 x_2) \mu_{x_1 \to f_b}, \quad (10)$$

where the first identity holds because $f_a(x_1)$ is an external factor node, the second identity is a result of Equation (5), and the third identity is a result of (6). Similarly, the messages that arrive to x_2 stating from the right of the graph are

$$\mu_{f_d, x_3} = f_d(x_3), \quad (11)$$

$$\mu_{x_3 \to f_c} = \mu_{f_c \to x_3} = f_d(x_3), \quad (12)$$

$$\mu_{f_c \to x_2} = \sum_{x_3} f_c(x_2, x_3) \mu_{x_3 \to f_c}, \quad (13)$$

where again the first identity results from the fact that $f_d(x_3, x_4)$ is an external factor node, the second results from Equation (5) and the third from Equation (6). Therefore, the marginal distribution of x_2 is now

$$p(x_2) \propto \mu_{f_b \to x_2} \times \mu_{f_c \to x_2}. \quad (14)$$

Using similar arguments we can derive $p(x_1)$ and $p(x_3)$.

In this particular example, the formula derived in (14) might seem redundant as for a wide class of distributions $p(\bullet)$, one can simply calculate the marginal distribution of x_2 using numerical integration. However, in high dimensions with many random variables, the sum-product rule can provide us with scalable and parallel posterior inference algorithms that can be several times faster compared to conventional algorithms that iterate sequentially (e.g., Gibbs sampler). It can be shown that the sum-product (belief propagation) algorithm is a special case of the more general expectation propagation algorithms that have been very popular in Bayesian machine learning; see Vehtari et al. (2018). Finally, note at this point that there is no mention about how to approximate the summations in (6), which will not necessarily be tractable. Given the sum-product formula, there are several algorithms that would allow for the approximation

[6] It is beyond the scope of this article to derive and prove the algorithm, and the reader is referred to the excellent machine learning books of Barber (2012) and Bishop (2006).

of the required messages which are functions of the factors f_j. For example, Wand (2017) developed message passing inference inspired by the variational Bayes method. In the next section, I adopt a recently developed algorithm (GAMP) that performs Normal approximations to the functions implied by the sum-product iterations.

3. Econometric Methodology

3.1. Time-Varying Parameter Regression

The starting point is the following TVP regression with stochastic volatility of the form

$$y_t = x_t\beta_t + \varepsilon_t, \tag{15}$$

subject to an initial condition for β_t at $t = 0$ (denoted as β_0), where y_t is the tth observation on the variable of interest, $t = 1,\ldots,T$, x_t is a $1 \times p$ vector of predictors (possibly including lags of y_t), β_t is a $p \times 1$ vector of coefficients, and $\varepsilon_t \sim N\left(0,\sigma_t^2\right)$ with σ_t^2 the time-varying variance parameter. It is desirable to estimate the initial condition in this model, rather than assume it is known. For that reason, following Frühwirth-Schnatter and Wagner (2010), this model can be written using an equivalent non-centered parameterization that allows to split the parameter β_t into a part that is constant (which is equivalent to its initial condition β_0), and an "add-on" time-varying part with initial condition fixed to zero. The equivalent specification is

$$y_t = x_t\widetilde{\beta} + x_t\widetilde{\beta}_t + \varepsilon_t, \tag{16}$$

where now $\widetilde{\beta}_t$ has initial condition zero and it holds that $\beta_t = \widetilde{\beta} + \widetilde{\beta}_t$. As shown in Belmonte, Koop, and Korobilis (2014) this parameterization allows to use shrinkage priors to determine whether a variable has constant coefficient (by only shrinking the time-varying part), or it is completely irrelevant for modeling y (by shrinking both the constant and time-varying parts to zero). More details of this approach are provided in the online appendix, Section D.1.

The TVP regression can be written in the following equivalent static regression form

$$y = \mathcal{X}\beta + \varepsilon, \tag{17}$$

where $y = \left[y_1,\ldots,y_T\right]'$ and $\varepsilon = [\varepsilon_1,\ldots,\varepsilon_T]'$ are column vectors stacking the observations y_t and ε_t respectively, $\beta = \left[\widetilde{\beta}', \widetilde{\beta}_1',\ldots,\widetilde{\beta}_T'\right]'$ is a $(T+1)p \times 1$ vector, and

$$\mathcal{X} = \begin{bmatrix} x_1 & x_1 & 0_{1\times p} & \ldots & 0_{1\times p} & 0_{1\times p} \\ x_2 & 0_{1\times p} & x_2 & \ldots & 0_{1\times p} & 0_{1\times p} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ x_{T-1} & 0_{1\times p} & 0_{1\times p} & \ldots & x_{T-1} & 0_{1\times p} \\ x_T & 0_{1\times p} & 0_{1\times p} & \ldots & 0_{1\times p} & x_T \end{bmatrix}, \tag{18}$$

is a $T \times (T+1)p$ matrix. It is evident that the first p columns of $\mathcal{X}$ specify a constant parameter regression and its remaining columns add "time-dummies" to that regression. The Gram matrix $\left(\mathcal{X}'\mathcal{X}\right)$ is of rank T and the $q = (T+1)p$, in total, regression coefficients in (17) cannot be estimated with OLS. For that reason, following a long-standing tradition in engineering, economists tend to assume that β_t (similarly for $\widetilde{\beta}_t$ in the non centered parameterization) typically follows a random walk of the form $\beta_t = \beta_{t-1} + \eta_t$, where $\eta_t \sim N(0,Q)$ for some $p \times p$ symmetric, positive-definite covariance matrix Q. This random walk regression for β_t allows to write the full TVP regression model in familiar state-space form, and also provides the additional information needed to estimate β_t using data y and $\mathcal{X}$. By doing so, estimation typically relies on MCMC methods by means of a simulation smoother; see Primiceri (2005) for a representative example. From a Bayesian point of view this additional information can be viewed as a conditional hierarchical prior of the form $p\left(\beta_t|\beta_{t-1}\right) \sim N\left(\beta_{t-1},Q\right)$ that provides appropriate level of shrinkage. Put differently, Equation (17) alone can be seen as an ill-posed problem where OLS does not have a unique solution and regularization is imperative for estimation.

In this article, I adopt this shrinkage view of the TVP regression model and propose an alternative inference strategy. That is, inference is done without reference to the useful but rather informative and subjective conditional hierarchical prior for β_t given β_{t-1} outlined above. Instead, the TVPs are recovered by estimating directly Equation (17) using data-based hierarchical shrinkage priors. In particular, I follow Tipping (2001) and define the following independent hierarchical prior for each element β_i of the vector β, $i = 1,2,\ldots,(T+1)p$,

$$p\left(\beta_i|\alpha_i\right) = N\left(0,\alpha_i^{-1}\right), \tag{19}$$

$$p\left(\alpha_i\right) = \text{Gamma}\left(\underline{a},\underline{b}\right). \tag{20}$$

This conditionally Normal prior for β_i and Gamma prior for the precision parameter α_i is a scale mixture of Normal representation of a Student's-t prior. Tipping (2001) calls this heavy-tailed prior a SBL prior, and I adopt this name henceforth; see also Korobilis (2013) for a detailed explanation why such hierarchical priors have good shrinkage properties. I follow Tipping (2001) and present all empirical results using the uniform hyperpriors (over a logarithmic scale) $\underline{a} = \underline{b} = 1 \times 10^{-10}$.

Two additional comments are in order regarding this TVP regression. First, the number of columns of $\mathcal{X}$ is $q = (T+1)p$, therefore, the number of coefficients grows rapidly. For example, with 700 monthly observations and only 100 predictors, we end up with 70,100 regression coefficients. As a consequence, it is imperative to choose a fast estimation algorithm that approximates the parameter posterior, and this is where the scalability of message passing algorithms comes into play. Second, there is no mention yet of inference on σ_t^2, as this issue is covered later in this section after the GAMP inference algorithm is outlined. In a nutshell, estimation of stochastic volatility σ_t^2 also follows the same shrinkage principles defined for β_t. That is, it is shown that we can write estimation of σ_t^2 as a high-dimensional regression problem, without having to assume any kind of first-order Markov dependence to σ_{t-1}^2.

3.2. A Factor Graph Representation of Bayesian Regression

At this point, we have all the necessary ingredients to cast the static form of the TVP regression in Equation (17) into a factor

graph form.[7] Consider first an independent (but not necessarily iid) prior for β, denoted $p(\beta) = \prod_{i=1}^{q} p(\beta_i)$, and the resulting posterior from Bayes theorem

$$p(\beta|y) \propto p(y|\beta)\, p(\beta), \tag{21}$$

$$= \prod_{t=1}^{T} p(y_t|\beta) \prod_{i=1}^{q} p(\beta_i). \tag{22}$$

The exact marginal posterior of β_i, $i = 1, \ldots, q$ is of the form

$$p(\beta_i|y) = \int p(\beta|y)\, d\beta_{j\neq i}, \tag{23}$$

$$\propto \int p(y|\beta)\, p(\beta)\, d\beta_{j\neq i}, \tag{24}$$

$$= p(\beta_i) \int p(y|\beta) \prod_{j=1, j\neq i}^{q} p(\beta_j)\, d\beta_{j\neq i}, \tag{25}$$

where $d\beta_{j\neq i}$ denotes integration over the whole set of $q-1$ parameters β_j for $j \neq i$. Therefore, the formula above requires integration over a $(q-1)$-dimensional integral, a numerical problem that can become computationally infeasible for a high-dimensional vector β.

We can now call the framework of factor graphs to factorize efficiently the marginal posteriors of β. The factor graph representation of the regression model is depicted in Figure 2. Based on this figure, the marginal posterior of β_i, presented in Equation (25), can be defined as the product of incoming messages at node β_i in the graph

$$p(\beta_i|y) = \mu_{p(\beta_i)\to\beta_i} \prod_{t=1}^{T} \mu_{p(y_t|\beta)\to\beta_i}. \tag{26}$$

Similar to Equation (8) in the example of Section 2, the message $\mu_{p(\beta_i)\to\beta_i}$ is an external factor node and for that reason it is equal to the prior $p(\beta_i)$. Generalizing the example sum-product rule derived in Equations (5) and (6) of the previous section, we can write the messages from $p(y_t|\beta)\ \forall t$ to β_i using the following expression

$$\mu_{p(y_t|\beta)\to\beta_i} = \int p(y_t|\beta) \prod_{j=1, j\neq i}^{p} \mu_{\beta_j\to p(y_t|\beta)} d\beta_{j\neq i}. \tag{27}$$

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

$$\mu_{\beta_j\to p(y_t|\beta)} = p(\beta_j) \prod_{s=1, s\neq t}^{T} \mu_{p(y_s|\beta)\to\beta_j}. \tag{28}$$

We can see in Equations (27) and (28) that 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 the following iterative sum-product scheme

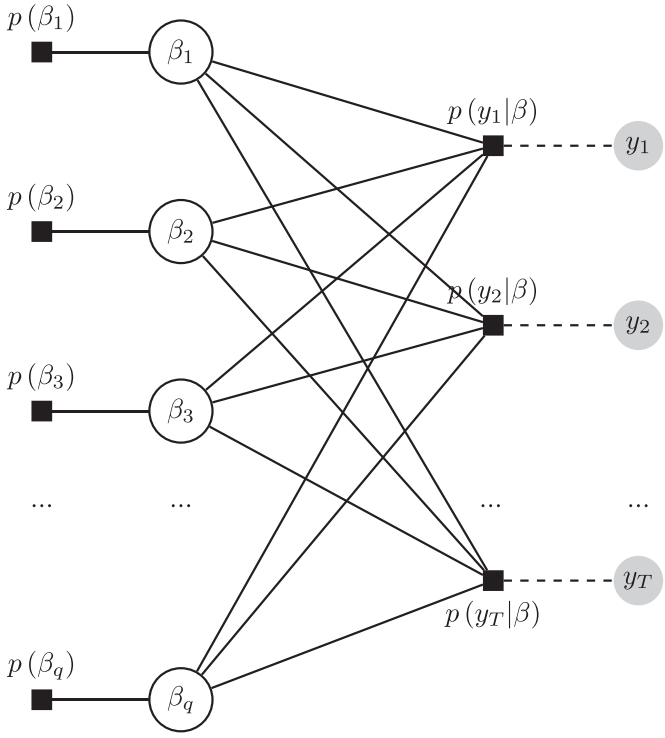

Figure 2. Factor graph representation for the high-dimensional regression model.

$$\mu^{(r+1)}_{p(y_t|\beta)\to\beta_i} = \int p(y_t|\beta) \prod_{j=1, j\neq i}^{q} \mu^{(r)}_{\beta_j\to p(y_t|\beta)} d\beta_{j\neq i}, \tag{29}$$

$$\mu^{(r+1)}_{\beta_j\to p(y_t|\beta)} = p(\beta_j) \prod_{f=1, f\neq t}^{T} \mu^{(r)}_{p(y_f|\beta)\to\beta_j}, \tag{30}$$

where the superscript (r) denotes the rth iteration of the algorithm. In graphs with a tree structure, one iteration of the algorithm above will always recover the exact marginal posteriors for the parameters β_i. In a factor graph with loops there are no guarantees that the sum-product rule will converge to a good fixed point. However, the sum-product rule can still achieve a good approximation and this is the reason why it is used extensively in applications of coding theory, machine vision, and compressive sensing that have a loopy graph representation (Mooij and Kappen 2007). Translating these facts into familiar jargon for the static regression in Equation (17), algorithmic convergence is achieved if the correlation of right-hand side predictors is not excessively high. If this is not the case, the joint posterior of the coefficients β might also be highly correlated, which would make inference solely based on the marginal posteriors $p(\beta_i)$ less accurate. In our benchmark TVP regression in (17), correlation is by default not excessively high due to the fact that the Gram matrix $\mathcal{X}'\mathcal{X}$ has a certain block-diagonal structure that allows for a general sparse correlation pattern—even if within a given block correlation may be high. In the empirical application, predictor variables are mainly principal components or lags thereof, such that correlation within each block is also low. Finally, note that the specific time-decomposition of the likelihood function does not accommodate autoregressive and general time-series models, where the likelihood at time t may

[7] For the sake of brevity, notation for prior, posterior, and likelihood distributions is generic, that is, there is no reference to their exact functional forms. Exact details and parametric formulas can be found in the online appendix, Section B.

be written conditional on past observations. In the empirical application it is found that, despite this approximation, autoregressive coefficients are recovered accurately.[8]

3.3. Generalized Approximate Message Passing

While the core of any message passing algorithm is fully described by the sum-product iterations, deriving the exact functional form of the messages in Equations (29) and (30) under the regression likelihood and the Student's-t hierarchical prior implies that cumbersome integrations might be necessary. The GAMP algorithm introduces certain Gaussian approximations to the sum-product iterations. Unlike Laplace approximations, that is, Gaussian approximations to parameter posteriors that many times can be poor, the GAMP approximation is fully based on asymptotic results that make it more reliable as the number of predictors grows large. First, when $q \to \infty$ a central limit theorem (CLT) postulates that the messages $\prod_{j=1,j\neq i}^{q} \mu_{\beta_j \to p(y_t|\beta)}$ can be approximated by a Gaussian distribution with respect to the uniform norm.[9] This result means that messages in (27) can be represented to be proportional to a Gaussian distribution. A second approximation involves taking the Taylor-series expansion of terms in the messages, so that the first two moments (mean and variance) of $p\left(\beta_i|y\right)$ can be obtained analytically up to the omission of $O\left(1/q\right)$ terms. Exact derivation of these approximations involves many tedious steps and transformations, and the reader is referred to the online appendix for more details. What is important to stress at this point is that both the CLT and Taylor-series approximations vanish as $q \to \infty$ with $q/T \to \delta$ for some constant δ; see Rangan (2011) and Rangan et al. (2016) for more details. This is an example of the "blessing of Big Data"—rather than the "curse of dimensionality" embedded in many traditional estimation algorithms—as the GAMP algorithm fully facilitates the large q asymptotics.

Deriving the GAMP algorithm involves several steps and lengthy proofs which are left for the online appendix. The final product of all the approximations to the two sum-product update equations (29) and (30) is a simple iterative algorithm that provides an approximation to the mean and variance of $p\left(\beta_i|y\right)$. The algorithm iterates through computationally trivial scalar multiplications and additions that result in worst case algorithmic complexity of $\mathcal{O}(Tq)$. That is, estimation of the marginal parameter posterior distribution does not involve costly operations such as high-dimensional integration or inversion of large matrices. This feature implies that the algorithm can handle regressions with an excessively large number of predictors with the same ease it can handle smaller regression models. Convergence is achieved when the difference between estimates of the posterior mean of β between two consecutive iterations is below a prespecified tolerance level. Other parameters can be updated by combining the GAMP algorithm with EM updates.[10] This feature is explained in the online appendix, where it is shown how to update the hyperparameter α_i introduced in the hierarchical prior of Equation (20).

Algorithm 1 Generalized approximate message passing (GAMP) with known variance and prior hyperparameters

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

A sketch of the algorithm is provided in Algorithm 1. This is a simplified version that focuses on estimation of β by assuming that the regression variance and prior hyperparameters are all known and fixed. Following the analysis in Section 2, the algorithm can be split into two steps: (i) evaluating all messages that leave each variable node β_j (output) and (ii) evaluating all messages that arrive at each variable node β_j (input). The final product is estimates of the posterior mean and variance of β_j which are denoted as $\widehat{\beta}_i$ and $\widehat{\tau}_i^{\beta}$, respectively. At the core of the calculation of the posterior mean and variance are the scalar functions g_{in} and g_{out}. Derivation of the exact form of these two functions depends on the form of the prior distribution and the likelihood. Online appendix, Section B, provides a

[8] A simulation exercise in the online appendix, Section C.3, generating artificial data from an AR(4) model, also verifies that the proposed GAMP algorithm performs well even if the likelihood function is not iid. Another assumption that affects performance of GAMP is that $\mathcal{X}$ is mean-zero Gaussian; see the discussion in Al-Shoukairi, Schniter, and Rao (2018) and references therein. In a time series context this means that GAMP will have better convergence when right hand-side predictors are strictly stationary, although the use of weakly stationary predictors is not excluded.

[9] This is a result of the Berry–Esseen CLT which states that a sum of random variables converge to a Gaussian density; see a proof of this theorem in Donoho, Maleki, and Montanari (2011). Given that the sum-product equations involve products of random variables, rather than sums, derivations of GAMP based on this CLT typically proceed by taking logarithms of Equations (26) and (28). The marginal posterior $p\left(\beta_i|y\right)$ is then recovered by performing an exponential transformation of the log messages, and by normalizing so that the posterior integrates to one; see the online appendix, Section A, for details.

[10] See Al-Shoukairi, Schniter, and Rao (2018) and Zou et al. (2016) for examples of how to derive EM updates for prior hyperparameters.

detailed algorithm in the case of the regression likelihood in Equation (17) and the prior in (19) and (20). In any case, Rangan (2011) showed that regardless of the form of the nonlinear scalar functions g_{in} and g_{out}, the worst-case complexity of the GAMP algorithm is not affected and is always $\mathcal{O}(Tq)$.

The algorithm above assumes a known regression variance, for example, normalized to be one. Of empirical interest is the derivation of an update rule for the variance parameter when this is both unknown and time varying. Here, I propose a novel, computationally trivial estimator of the variance that builds on approximations used in the Bayesian stochastic volatility estimator of Kim et al. (1998). First, we write the regression model in (17) in the following form

$$y = \mathcal{X}\beta + \Sigma\nu, \tag{31}$$

where Σ is a $T \times T$ diagonal matrix with the time-varying standard deviations σ_t on its main diagonal. Subsequently, conditional on knowing β by means of some estimate $\widehat{\beta}$, we can rewrite the above model as

$$\log\left[\left(y - \mathcal{X}\widehat{\beta}\right)^2\right] = \log\left(\text{diag}\left(\Sigma\right)^2\right) + \log(\nu^2), \Rightarrow \tag{32}$$

$$\widetilde{y} = \widetilde{\sigma}^2 + \widetilde{\nu}, \tag{33}$$

where $\text{diag}\left(\Sigma\right)^2$ is a $T \times 1$ vector with elements σ_t^2 $\forall t \in [1, T]$, and variables with a $\widetilde{\bullet}$ denote quantities in log-squares. In particular, the distribution of $\widetilde{\nu}$ is $\log - \chi^2$ with one degree of freedom. Following Kim et al. (1998), we can approximate this with a mixture of seven Normal distributions with means μ_i, variances V_i and component weights π_i, where $i = 1, \ldots, 7$ and $\sum_i \pi_i = 1$.[11] Then Equation (33) can be replaced with the following set of seven equations

$$\widetilde{y} = \widetilde{\sigma}^2 + u_i, \quad i = 1, \ldots, 7, \tag{34}$$

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

$$\widehat{\sigma}_t^2 = \exp\left(\sum_{i=1}^{7} \pi_i \left(\widetilde{y}_t - \mu_i\right)/7\right). \tag{35}$$

Similar expressions can also be derived for the posterior variance of σ_t^2 if desired, for example, when computing the posterior predictive density via simulation. It turns out that the resulting estimate of volatility is similar to the standard stochastic volatility estimator of Kim et al. (1998), but it is much less persistent due to the lack of dependence of σ_t^2 on σ_{t-1}^2. More evidence on the excellent properties of this simple estimator of stochastic volatility is provided in the online appendix, Section D.1.

Finally, online appendix, Section C, provides detailed Monte Carlo evidence on the usefulness of the proposed econometric specification and algorithm. By simulating artificial data from models with various patterns of time-variation in parameters, it is assessed how good the specification in Equation (17), with the assistance of the SBL prior, is at recovering the true TVPs. At the same time, a second simulation exercise shows the ability of the GAMP algorithm with shrinkage prior to perform high-dimensional shrinkage even in cases with more predictors than observations. A final simulation exercise discusses the stability of the GAMP algorithm in models with correlated predictors, and assesses numerically the case where the likelihood function is not iid. While the results of the simulated data exercises suggest that the proposed algorithm provides a reasonable balance between computational speed and estimation accuracy, the next section establishes that the proposed algorithm is also very useful in a forecasting application using real macroeconomic data.

[11] The exact values of μ_i, V_i, π_i for all seven components is provided in the online appendix, Section B.

4. Empirical Illustration: Forecasting Inflation

This section describes the setup and results of a comprehensive forecasting exercise that demonstrates the merits of the modeling approach outlined in the previous section. Most applications of TVP regressions focus in particular on inflation. Of course, this class of models is flexible enough to provide useful forecasts of any other variable of interest; see Bauwens et al. (2015) for assessing structural breaks in several monthly and quarterly macroeconomic time series. Nevertheless, there is ample evidence that structural breaks in inflation are so evident and complex, such that TVP models are particularly useful for forecasting this variable (see Chan et al. 2012; Koop and Korobilis 2012; Groen, Paap, and Ravazzollo 2013; Pettenuzzo and Timmermann 2017; Stock and Watson 2007, among many others).

The data collected for this exercise are 115 macroeconomic variables from Federal Reserve Economic Data (FRED) of St. Louis Federal Reserve Bank website. The data originally span the period 1959M1 to 2016M6, but the effective sample is smaller after taking stationarity transformations and lags. The stationarity transformations follow standard norms in this literature (see Stock and Watson 1999) and exact details are provided in the online appendix, Section A.

The empirical application builds on the seminal work of Stock and Watson (1999) for forecasting inflation. These authors specify the following benchmark forecasting model

$$\pi_{t+h}^h - \pi_t = \phi_0 + z_t\theta(L) + \Delta\pi_t\gamma(L) + e_{t+h}, \tag{36}$$

where $\pi_t^h = (1200/h)\log\left(P_t/P_{t-h}\right)$ is the h-period inflation in the price level P_t. As Stock and Watson (1999, sec. 2) explain in detail the assumption here is that inflation is $I(1)$ while the exogenous variables in z_t are $I(0)$. Two modifications of this basic forecasting model are in order. First, as Stock and Watson (1999) also suggest, the high-dimensional variables z_t are replaced by factors f_t estimated using principal components. Second, the forecasting equation is enhanced with TVPs and stochastic volatility. The final forecasting model used in this article is of the form

$$\pi_{t+h}^h - \pi_t = \phi_{t,0} + f_t\theta_t(L) + \Delta\pi_t\gamma_t(L) + e_{t+h}, \tag{37}$$

where $e_t \sim N\left(0, \sigma_t^2\right)$ and f_t is a lower-dimensional vector of factors.

The forecasting exercise is run for two measures of inflation, namely the consumer price index for all items (CPIAUCSL) and the personal consumption expenditures price index (PCEPI). The forecast horizons evaluated are $h = 1, 3, 6, 12$ which

correspond to one-month, one-quarter, one-semester, and one-year ahead forecasts, respectively. Following Bauwens et al. (2015) evaluation of forecasts is based on the mean square forecast error (MSFE) for point forecasts, and on the logarithm of the average predictive likelihoods (log APL) for comparing whole forecast densities. Exactly 50% of the sample is used for evaluation of out-of-sample forecasts, leading to a period of $343 - h$ months where MSFEs and log APLs are calculated. Note that while estimation entails the spread $\pi_{t+h}^h - \pi_t$, all forecast evaluations in this section (see also alternative model in Equation (38)) pertain to π_{t+h}^h.

When applying the proposed GAMP estimation methodology, Equation (37) is estimated using two own lags of the dependent variable, the first 20 principal component estimates of the factors f_t (updated recursively using only information up to time t) and two lags of these factors (i.e., their values in periods t and $t-1$). As explained in the main text, this TVP model can be estimated using GAMP by casting it into the form (15) by setting $y_t = \pi_{t+h}^h - \pi_t$, $x_t = \left[1, f_t, \Delta\pi_t\right]$, $\beta_t = \left(\phi_{t,0}, \theta_t(L)', \gamma_t(L)'\right)'$ and $e_{t+h} = \varepsilon_t$. Written in this static form and using all available observations, the proposed empirical model has nearly 30,000 regression coefficients and another 700 volatility parameters to estimate. The only input that the GAMP algorithm requires is choice of two scalar prior hyperparameters. For the SBL prior of Equations (19) and (20) these hyperparameters are set, as explained in Section 3, to the uniform values $\underline{a} = \underline{b} = 1 \times 10^{-10}$. This approach to estimating the TVP regression of (37) using GAMP is abbreviated as TVP-GAMP in the results presented next.

The benchmark time-varying regression approach estimated with the GAMP algorithm is contrasted against a range of popular algorithms for inference in models with many predictors and/or stochastic variation in coefficients. The list of competing specifications and estimation algorithms is the following:

- KP-AR: This is a structural breaks AR(2) model based on Koop and Potter (2007). It only features an intercept and two lags of inflation.
- GK-AR: This is a structural breaks AR(2) model based on Giordani and Kohn (2008). It only features an intercept and two lags of inflation.
- TVP-AR: This is a typical TVP-AR(2) model with stochastic volatility, estimated with MCMC methods, similar to Pettenuzzo and Timmermann (2017). It only features an intercept and two lags of inflation.
- UCSV: The unobserved components stochastic volatility model of Stock and Watson (2007) is a special case of a TVP regression with no predictors—it is a local level state-space model featuring stochastic volatility in the state equation.
- TVD: The time-varying dimension (TVD) model of Chan et al. (2012) features an intercept, two lags of inflation, and the first three principal components estimates of the factors. The number of factors is restricted to three for computational reasons. Also for computational reasons one cannot do time-varying selection among all possible 2^p models constructed with p predictors, therefore, I follow Chan et al. (2012) and do dynamic selection of either models with one variable at a time, or the full model with all variables.
- TVS: The time-varying shrinkage (TVS) algorithm of Kalli and Griffin (2014) features an intercept, two lags and the first three principal components estimates of the factors (also restricted to three factors for computational reasons).
- TVP-BMA: Introducing a Bayesian model averaging prior in the TVP regression is fairly trivial as Groen, Paap, and Ravazzollo (2013) have shown. We can use with this algorithm up to 10 principal component estimates of the factors, an intercept and two lags of inflation.
- BMA: This is a constant parameter version of the forecasting regression specification that features the stochastic search variable selection (SSVS) of George and McCulloch (1993). Even though this prior can be also used for variable selection, here it is used in a Bayesian model averaging (BMA) setting. For this algorithm we use the same number of predictors as in TVP-GAMP, namely an intercept, two own lags of inflation, and two lags of the first 20 principal components. However, this model is the only one in the comparison that does not have TVPs.

All these models collapse to being special cases of the benchmark equation (37), despite the fact that different specifications might imply various additional assumptions about how the coefficients might evolve over time (whereas TVP-GAMP does not rely on such additional assumptions). All models except for the UCSV have in common an intercept and the two own lags of inflation.[12] For those algorithms that rely on shrinkage priors (TVP-GAMP, TVD, TVS, TVP-BMA, and BMA) the intercept and the two lags of inflation are never allowed to shrink by using a noninformative prior on them. Therefore, whenever shrinkage (static or dynamic) is implemented this only applies to the exogenous information in the factors. Exact details of the econometric specifications and prior settings associated with the competing models is provided in the online appendix, Section E.

A final note is on computation. All of the competing models listed above are based on estimation using MCMC and in particular the Gibbs sampler. Most of these models were originally developed by their respective authors for forecasting inflation. This is due to the fact that time-varying parameter regressions have consistently been found to be superior for this series. However, even though one would normally expect more breaks to be present in higher frequency monthly inflation, all of these articles estimate their models using quarterly data. This is done for computational reasons. Due to the fact that here these models are estimated for monthly data, I follow Bauwens et al. (2015)

[12] To understand better whether forecast gains can be achieved from specifying a model with many predictors, or with flexible time-variation, or both, I only calculate direct multistep forecasts from all competing models. That way all algorithms are used to estimate different versions of the same regression with y_{t+h} on the left hand side (for each h) and information dated t or earlier on the left hand side. However, iterated forecasts can be computed from models with no exogenous predictors (e.g., TVP-AR or UCSV). Direct forecasts are better when the model is misspecified, while iterated forecasting models in general result in more efficient econometric estimates and sharper predictive densities. Examination of $h = 12$ month ahead iterated forecasts from the KP-AR, GK-AR, TVP-AR, and UCSV models reveals that these are, most times, slightly inferior to respective direct forecasts in terms of MSFE, but they can be in some cases up to 15% better in terms of average log predictive likelihoods. Iterated forecasting results are not presented here, but they are available from the author.

Table 1. Point forecast performance: MSFEs relative to AR(2) benchmark.

	CPI				PCE deflator			
	$h = 1$	$h = 3$	$h = 6$	$h = 12$	$h = 1$	$h = 3$	$h = 6$	$h = 12$
KP-AR	0.952	0.998	0.978	0.904	1.030	1.027	1.042	0.973
GK-AR	0.997	1.007	1.002	0.993	1.005	1.006	0.999	0.997
TVP-AR	1.009*	1.047***	1.258***	1.224***	1.053***	1.119***	1.141***	1.123***
UCSV	1.032**	1.063***	1.286**	1.312	1.060**	1.077**	1.234*	1.159***
TVD	1.014	0.988	0.942	0.919	1.091	1.108	1.156	1.029
TVS	1.146**	1.547	1.555	1.155	1.084*	1.413	2.150	1.208*
BMA	0.965***	0.952***	0.883***	0.859***	0.993	0.971	0.94**	0.929***
TVP-BMA	1.089	0.981	1.099	0.825	1.268	1.198	1.490	1.091
TVP-GAMP	0.988*	0.870***	0.749***	0.714***	1.020	0.965	0.915***	0.866***

NOTE: Model acronyms are as follows: KP-AR: Koop and Potter (2007) structural breaks AR(p) model; GK-AR: Giordani and Kohn (2008) structural breaks AR(p) model; TVP-AR: Pettenuzzo and Timmermann (2017) time-varying parameter AR(p) model; UCSV: Stock and Watson (2007) unobserved components stochastic volatility; TVD: Chan et al. (2012) time-varying dimension regression; TVS: Kalli and Griffin (2014) time-varying sparsity regression; BMA: George and McCulloch (1993) stochastic search variable selection regression; TVP-BMA: Groen, Paap, and Ravazzollo (2013) time-varying Bayesian model averaging model; TVP-GAMP: Shrinkage representation of time-varying parameter regression, with generalized approximate message passing estimation.
Next to MSFE values the results of the Diebold–Mariano statistic are presented, with
*Significance at the 10% level.
**Significance at the 5% level.
***Significance at the 1% level.

Table 2. Density forecast performance: log APLs relative to AR(2) benchmark.

	CPI				PCE deflator			
	$h = 1$	$h = 3$	$h = 6$	$h = 12$	$h = 1$	$h = 3$	$h = 6$	$h = 12$
KP-AR	0.090	0.081	0.002	−0.036	0.011	0.074	−0.057	−0.035
GK-AR	−0.025	−0.029	0.004	0.037	0.034	0.138	0.056	0.052
TVP-AR	0.118	0.111	0.181	0.067	0.036	0.007	−0.036	−0.029
UCSV	0.161	0.239	0.224	0.144	0.048	0.245	−0.067	0.059
TVD	−0.103	−0.005	−0.339	−0.380	−0.097	0.062	−0.885	−0.262
TVS	0.018	−0.163	−0.660	−0.367	−0.001	0.003	−0.427	−0.246
BMA	0.030	−0.067	0.042	0.084	−0.056	−0.002	−0.062	0.030
TVP-BMA	0.121	0.313	0.413	0.399	−0.026	0.227	0.205	0.219
TVP-GAMP	−0.204	0.258	0.320	0.321	0.061	0.260	0.045	0.191

NOTE: See notes in Table 1 for details of model acronyms.

and base inference only on 5000 samples from the posterior after a burn-in period of 1000 draws, that is, a total of 6000 MCMC iterations. Convergence criteria suggest that such low number of iterations is sufficient for forecasting, even though it might not be satisfactory for other econometric exercises. Despite the low number of MCMC iterations, computation is quite cumbersome taking several hours for some models. In contrast, it takes only minutes to run the full recursive exercise using the TVP-GAMP model that features both time-varying parameters and the full set of available predictors. The GAMP algorithm not only involves simple scalar computations, but also converges fairly quickly after 10–100 iterations. Once convergence is achieved, the first two posterior moments are readily available for further inference, rather than having to store thousands of samples from the posterior of a high-dimensional parameter vector.

The results from this forecasting exercise are presented in Tables 1 and 2, and are very encouraging for the proposed TVP-GAMP method. Table 1 shows MSFEs relative to an AR(2) benchmark (with an intercept), such that numbers lower than one signify better performance of a competing model relative to that benchmark AR(2) specification. It can be seen that under the specified regression model, point forecasts from TVP-GAMP dominate alternatives by a substantial amount, both for CPI and PCE inflation. The forecast gains are increasing with the horizon. Table 2 shows the logarithm of the average predictive likelihood (log APL), and this metric is quoted as a spread from the log APL of the simple AR(2) specification. Positive values signify better performance relative to the benchmark AR(2). Using this metric, TVP-GAMP is either the top performing model or among the top, for the four forecast horizons and the two measures of inflation.

It is notable that these results contradict the previous claims that time-variation in parameters is important for inflation. The three models with the largest number of predictors, namely BMA and TVP-GAMP, and to a lesser degree TVP-BMA, seem to be improving a lot over time-varying parameter models with no predictors. The results seem to suggest that information in predictors is more important than the specification of time variation in regression parameters. This observation is not undermined by the fact that point forecasts from TVP-BMA are not significant, and that density forecasts from BMA are quite poor relative to TVP-BMA and TVP-GAMP. First, TVP-BMA is overparameterized[13] its point forecast performance is not as good as the more conservative (in terms of time-variation in parameters, not available number of predictors) BMA and TVP-GAMP specifications. Second, when considering density forecasts, BMA is definitely misspecified since it does not allow for stochastic volatility, and it naturally does not perform as

[13] Shrinkage in TVP-BMA is only across predictors, but this model does not restrict the amount of time-variation in parameters.

Table 3. Point and density forecast performance using alternative definition of the CPI forecasting regression.

	MSFE				log APL			
	$h=1$	$h=3$	$h=6$	$h=12$	$h=1$	$h=3$	$h=6$	$h=12$
KP-AR	0.901	0.706	0.756	0.544	0.042	0.321	0.154	0.128
GK-AR	0.963	0.929	0.900	0.882	0.071	0.168	0.014	0.125
TVP-AR	0.852	0.917	0.800	0.587	0.210	0.353	0.422	0.114
UCSV	0.911	0.898	0.817	0.638	0.114	0.163	0.118	0.154
TVD	0.902	0.851	0.863	0.873	−0.041	0.150	0.022	0.021
TVS	0.960	0.929	0.891	0.905	0.033	0.134	0.033	0.037
BMA	0.995	1.109	1.233	0.914	0.118	0.273	0.151	0.187
TVP-BMA	0.926	0.903	0.805	0.650	0.092	0.088	0.087	0.126
TVP-GAMP	0.944	0.876	0.819	0.768	0.190	0.276	0.264	0.136

NOTE: See notes in Table 1 for details of model acronyms. Unlike the previous two tables that present results for both CPI and PCE, this table only shows results for CPI, where its left panel focuses on MSFEs and its right panel on log APLs. However, as in the previous two tables, MSFEs and log APLs are relative to an AR(2) benchmark. MSFE entries lower than one mean that the estimation method of the respective row does better than the benchmark. Log APL entries higher than zero mean that the estimation method of the respective row does better than the benchmark.

well as TVP-BMA and TVP-GAMP that allow for changing variance. Therefore, these findings suggest that TVP-GAMP is overall the best model and that its specification is flexible enough to capture both structural change and use information in a large set of predictors at the same time. Most importantly, the SBL prior allows to strike a good balance between these two modeling characteristics by removing irrelevant predictors as well as regularizing time variation.

These results are in stark contrast to existing results for TVP models presented in the articles cited above (see, e.g., footnotes in Table 1). The culprit is simply the assumption that inflation is I(1) that Stock and Watson (1999) introduced in their seminal article, and that it is adopted in Equation (37). Once the random walk dynamics are removed from inflation (i.e., inflation gap becomes the dependent variable), the role of time-varying parameters in forecasting becomes less important and the most significant feature is the information included in exogenous predictors. It would be interesting then, as a robustness check, to specify the forecasting regression for inflation using the following form

$$\pi_{t+h}^{h} = \phi_{t,0} + f_t\theta_t(L) + \pi_t\mu_t(L) + e_{t+h}. \tag{38}$$

This equation is more in line with the forecasting model estimated in articles such as Chan et al. (2012), Groen, Paap, and Ravazzollo (2013), or Pettenuzzo and Timmermann (2017).

Table 3 shows results based on this alternative specification of Equation (38) for CPI inflation only. The left part of the table presents MSFE results, while the right panel presents log APLs. In this case, it is evident that the various variants of TVP models considered improve tremendously over the benchmark. As a matter of fact, models such as the KP-AR, TVP-AR, and UCSV also improve a lot relative to the constant parameter BMA. Looking at point forecasts and the associated MSFE results, we can observe many differences among TVP models, especially as the forecast horizon increases. For example, the structural breaks KP-AR specification has the lowest relative MSFE for $h = 12$ among all models, but also the structural breaks GK-AR specification is among the worst performing (but still much better than the simple AR model). TVD and TVS estimated with the monthly data are not only cumbersome, but also do not perform as well as TVP models with no predictors. In contrast, the TVP-BMA algorithm is performing quite well, even though it still does not beat TVP models with no predictors. In this alternative forecasting regression, TVP-GAMP is not the top forecasting model but its performance is still quite good. If it was not for the exceptional performance of the KP-AR model, TVP-GAMP would have been a top model for $h = 1, 3, 6$.

When looking at density forecast evaluation the results might not comply with the results for the point forecasts. Still good performing models are the KP-AR and the TVP-AR, but now the BMA and TVP-GAMP beat models such as the UCSV. With such diverse set of flexible models it is hard to pin down which exact features help in point and density forecasts. Nevertheless, for the forecasting regression (38) it seems that the way time variation in parameters is specified is more important than information in exogenous predictors. Further numerical evidence on the relative forecast performance of some of the competing models, is provided in the online appendix, Section D.2.

5. Conclusions

This article evaluates a new methodology for performing Bayesian inference in high-dimensional regression models. The proposed GAMP is a fast algorithm for approximating iteratively the first two moments of the marginal posterior distribution of a high-dimensional vector of coefficients. It is established how effortlessly the GAMP algorithm can be extended with interesting modeling features such as hierarchical shrinkage priors, time-varying coefficients and stochastic volatility, and many predictors. The benefit of the proposed approach is demonstrated using an inflation forecasting exercise that leads to the recursive estimation of regression models with thousands of covariates. Due to the low algorithmic complexity, GAMP could be generalized to much higher dimensions with millions of predictors/covariates, as it is also trivially parallelizable.

The current study opens up new avenues for research. First, the proposed framework for modeling time-varying parameters using hierarchical shrinkage priors can be extended in interesting ways. For example, shrinkage estimators/priors that apply on group of coefficients (such as the Group Lasso) can be used in this setting so that coefficients are shrunk

either in groups of predictors for a given time period or in groups of consecutive time periods for a given predictor. This is because in the TVP setting the vector of regression coefficients β has elements that correspond both to predictor j, $j = 1, \ldots, p$, but also to time period t, $t = 1, \ldots, T$. One can think of other shrinkage priors in order to perform a more structured approach to uncovering patterns of time-variation in parameters, such as various pooling priors used in the panel data literature. Finally, the article proposes the framework of factor graphs for designing efficient algorithms. Many macroeconomic problems currently do not typically involve extensive use of Big Data sets, however, they involve multivariate models with possibly thousands of coefficients, such as VAR, factor, and DSGE models. Bayesian estimation of these models is quite cumbersome, many times relying on linear or nonlinear state-space methods. As empirical macroeconomic models become larger and more complex, factor graph inference could help economists come up with novel efficient algorithms and unveil new features in macroeconomic data.

Supplementary Materials

This article is accompanied by an Online Appendix and replication codes for all the quantitative results presented in the main article and the Appendix. The Online Appendix includes description of data; proofs, derivations, and exact computational details of the algorithm proposed in the article; extensive Monte Carlo comparisons; and further empirical results. The accompanying replication files are written in MATLAB, and they can also be accessed on the Author's personal website, *https://sites.google.com/site/dimitriskorobilis/*.

Acknowledgments

This article has significantly improved based on suggestions from two anonymous referees, an associate editor, and the editor (Todd Clark), whom I gratefully acknowledge. I would like to thank Fabio Canova, Eric Ghysels, George Kapetanios, Gary Koop, Massimiliano Marcellino, Geert Mesters, Davide Pettenuzzo, Giorgio Primiceri, Giuseppe Ragusa, Giovanni Ricco, Lucrezia Reichlin, Frank Schorfheide, Herman van Dijk, Hal Varian, and Mike West for useful comments and/or stimulating discussions. I would also like to thank Joshua Chan and Maria Kalli/Jim Griffin for sharing MATLAB codes that replicate time-varying parameter specifications suggested by these authors. Additionally, I would like to acknowledge helpful comments and questions from participants at the 2017 Deutsche Bundesbank Forecasting Workshop; the 2017 Norges Bank conference on "Big data, Machine Learning and the Macroeconomy"; the 10th ECB Workshop on Forecasting Techniques: "Economic Forecasting with Large Datasets"; the BayesComp/ISBA 2018 conference; the 3rd annual Now-Casting.com Workskhop; the 2018 Budapest School for Central Bank Studies; the 2019 Econometric Institute workshop "Machine Learning Meets Econometrics"; and the 2019 Joint Research Centre European Commission workshop "Big Data and Economic Forecasting." Finally, I would like to acknowledge useful feedback from seminar participants at various universities and central banks.

References

Al-Shoukairi, M., Schniter, P., and Rao, B. D. (2018), "A GAMP-Based Low Complexity Sparse Bayesian Learning Algorithm," *IEEE Transactions on Signal Processing*, 66, 294–308. [6]

Angelino, E., Johnson, M. J., and Adams, R. P. (2016), "Patterns of Scalable Bayesian Inference," *Foundations and Trends® in Machine Learning*, 9, 119–247. [1]

Barber, D. (2012), *Bayesian Reasoning and Machine Learning*, New York: Cambridge University Press. [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, 596–620. [7,8]

Belmonte, M., Koop, G., and Korobilis, D. (2014), "Hierarchical Shrinkage in Time-Varying Coefficients Models," *Journal of Forecasting*, 33, 80–94. [2,4]

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, New York: Springer. [3]

Chan, J., Koop, G., Leon-Gonzalez, R., and Strachan, R. (2012), "Time Varying Dimension Models," *Journal of Business and Economic Statistics*, 30, 358–367. [2,7,8,9,10]

Cooley, T. F., and Prescott, E. C. (1976), "Estimation in the Presence of Stochastic Parameter Variation," *Econometrica*, 44, 167–184. [2]

Donoho, D. L., Maleki, A., and Montanari, A. (2009), "Message Passing Algorithms for Compressed Sensing," *Proceedings of National Academy of Sciences of the United States of America*, 106, 18914–18919. [1]

——— (2011), "How to Design Message Passing Algorithms for Compressed Sensing," unpublished manuscript, available at *http://www.ece.rice.edu/~mam15/bpist.pdf*. [6]

Frühwirth-Schnatter, S., and Wagner, H. (2010), "Stochastic Model Specification Search for Gaussian and Partial Non-Gaussian State Space Models," *Journal of Econometrics*, 154, 85–100. [4]

George, E. I., and McCulloch, R. E. (1993), "Variable Selection via Gibbs Sampling," *Journal of the American Statistical Association*, 88, 881–889. [8,9]

Giordani, P., and Kohn, R. (2008), "Efficient Bayesian Inference for Multiple Change-Point and Mixture Innovation Models," *Journal of Business and Economic Statistics*, 26, 66–77. [2,8,9]

Groen, J. J. J., Paap, R., and Ravazzollo, F. (2013), "Real Time Inflation Forecasting in a Changing World," *Journal of Business and Economic Statistics*, 31, 29–44. [2,7,8,9,10]

Herbst, E. P., and Schorfheide, F. (2015), *Bayesian Estimation of DSGE Models*, Princeton, NJ: Princeton University Press. [1]

Jackson, M. O. (2008), *Social and Economic Networks*, Princeton, NJ: Princeton University Press. [1]

Kalli, M., and Griffin, J. E. (2014), "Time-Varying Sparsity in Dynamic Regression Models," *Journal of Econometrics*, 178, 779–793. [2,8,9]

Kim, S., Shephard, N., and Chib, S. (1998), "Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models," *The Review of Economic Studies*, 65, 361–393. [7]

Koop, G., and Korobilis, D. (2010), "Bayesian Multivariate Time Series Methods for Empirical Macroeconomics," *Foundations and Trends® in Econometrics*, 3, 267–358. [1]

——— (2012), "Forecasting Inflation using Dynamic Model Averaging," *International Economic Review*, 53, 867–886. [7]

Koop, G., and Potter, S. M. (2007), "Estimation and Forecasting in Models with Multiple Breaks," *The Review of Economic Studies*, 74(3), 763–789. [2,8,9]

Korobilis, D. (2013), "Hierarchical Shrinkage Priors for Dynamic Regressions With Many Predictors," *International Journal of Forecasting*, 29, 43–59. [4]

Kowal, D. R., Matteson, D. S., and Ruppert, D. (2019), "Dynamic Shrinkage Processes," *Journal of the Royal Statistical Society Series B*, 81, 781–804. [2]

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. [1]

Mooij, J., and Kappen, H. (2007), "Sufficient Conditions for Convergence of the Sum-Product Algorithm," *IEEE Transactions on Information Theory*, 53, 4422–4437. [5]

Nakajima, J., and West, M. (2013), "Bayesian Analysis of Latent Threshold Dynamic Models," *Journal of Business and Economic Statistics*, 31, 151–164. [2]

Pearl, J. (1982), "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach," in *Second National Conference on Artificial Intelligence, AAAI-82*, Pittsburgh, PA, Menlo Park, CA: AAAI Press, pp. 133–136. [3]

Pettenuzzo, D., and Timmermann, A. (2017), "Forecasting Macroeconomic Variables Under Model Instability," *Journal of Business and Economic Statistics*, 35, 183–201. [7,8,9,10]

Primiceri, G. E. (2005), "Time Varying Structural Vector Autoregressions and Monetary Policy," *The Review of Economic Studies*, 72(3), 821–852. [4]

Rangan, S. (2011), "Generalized Approximate Message Passing for Estimation With Random Linear Mixing," *IEEE International Symposium on Information Theory*, pp. 2174–2178. [1,6,7]

Rangan, S., Schniter, P., Riegler, E., Fletcher, A. K., and Cevher, V. (2016), "Fixed Points of Generalized Approximate Message Passing With arbitrary Matrices," arXiv no. 1301.6295v4. [6]

Ročková, V., and McAlinn, K. (2018), "Dynamic Variable Selection With Spike-and-Slab Process Priors," Technical Report, Booth School of Business, University of Chicago. [2]

Stock, J. H., and Watson, M. W. (1999), "Forecasting Inflation," *Journal of Monetary Economics*, 44, 293–335. [2,7,10]

——— (2007), "Why Has U.S. Inflation Become Harder to Forecast?" *Journal of Money, Credit and Banking*, 39, 3–33. [2,7,8,9]

Tipping, M. E. (2001), "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning Research*, 1, 211–244. [2,4]

Vehtari, A., Gelman, A., Sivula, T., Jylänki, P., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D., and Robert, C. (2018), "Expectation Propagation as a Way of Life: A Framework for Bayesian Inference on Partitioned Data," arXiv no. 1412.4869v3. [3]

Wand, M. P. (2017), "Fast Approximate Inference for Arbitrarily Large Semiparametric Regression Models via Message Passing," *Journal of the American Statistical Association*, 112, 137–168. [1,3,4]

Zou, X., Li, F., Fang, J., and Li, H. (2016), "Computationally Efficient Sparse Bayesian Learning via Generalized Approximate Message Passing," in *IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pp. 1–4. [6]