

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Helske, Jouni

Title: Efficient Bayesian generalized linear models with time-varying coefficients : The walker package in R

Year: 2022

Version: Published version

Copyright: © 2022 The Author(s). Published by Elsevier B.V.

Rights: CC BY 4.0

Rights url: <https://creativecommons.org/licenses/by/4.0/>

Please cite the original version:

Helske, J. (2022). Efficient Bayesian generalized linear models with time-varying coefficients : The walker package in R. *SoftwareX*, 18, Article 101016.
<https://doi.org/10.1016/j.softx.2022.101016>



Original software publication

Efficient Bayesian generalized linear models with time-varying coefficients: The walker package in R

Jouni Helske

Department of Mathematics and Statistics, University of Jyväskylä, FI-40014 Jyväskylä, Finland



ARTICLE INFO

Article history:

Received 15 May 2020

Received in revised form 10 August 2020

Accepted 11 February 2022

Keywords:

Bayesian inference

Time-varying regression

R

Markov chain Monte Carlo

ABSTRACT

The R package walker extends standard Bayesian general linear models to the case where the effects of the explanatory variables can vary in time. This allows, for example, to model the effects of interventions such as changes in tax policy which gradually increases their effect over time. The Markov chain Monte Carlo algorithms powering the Bayesian inference are based on Hamiltonian Monte Carlo provided by Stan software, using a state space representation of the model to marginalize over the regression coefficients for efficient low-dimensional sampling.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Code metadata

Current code version
Permanent link to code/repository used for this code version
Code Ocean compute capsule
Legal Code License
Code versioning system used
Software code languages, tools, and services used
Compilation requirements, operating environments & dependencies

1.0.3-1
https://github.com/ElsevierSoftwareX/SOFTX_2020_212
none
GPL3
git
R, Stan, C++.
R version 3.4.0 and up, C++14, R packages bayesplot, BH, coda, dplyr, ggplot2, Hmisc, KFAS, loo, methods, RcppParallel, rlang, rstan, rstantools, StanHeaders, Rcpp, RcppArmadillo, RcppEigen
<https://cran.r-project.org/web/packages/walker/walker.pdf>
jouni.helske@iki.fi

If available Link to developer documentation/manual
Support email for questions

1. Motivation and significance

Assume a time series of interest y_1, \dots, y_T which is linearly dependent on the some other predictor time series X_1, \dots, X_T , where $X_t' = (x_{1t}, \dots, x_{pt})'$ are the predictor variables at the time point t , and define linear-gaussian time series regression model as

$$y_t = X_t' \beta_t + \epsilon_t, \quad t = 1, \dots, T, \quad (1)$$

where $\epsilon_t \sim N(0, \sigma_\epsilon^2)$ and β is a vector of p unknown regression coefficients (first one typically being the intercept term).

It is not always reasonable to assume that the relationship between y_t and some predictor x_{it} stays constant over $t = 1, \dots, T$, the time period of interest. The convenient linear relationship approximation may hold well for only piecewise if the underlying relationship is nonlinear or when there are some unmeasured

confounders that alter the relationship between the measured variables. Allowing the regression coefficients to vary over time can in some instances alleviate these problems. It is also possible that our knowledge of the phenomena of interest already leads to suspect time-varying relationships (see, e.g., Chapter 1 in [1]).

The basic time series regression model (1) can be extended to allow the unknown regression coefficients β_t to vary over time. This can be done in various ways, for example, by constructing a semiparametric model based on kernel smoothing [2], or parametrically by using dynamic Bayesian networks [3] or state space models [4]. We follow the state space modelling approach and define a gaussian time series regression model with random walk coefficients as

$$\begin{aligned} y_t &= X_t' \beta_t + \epsilon_t, \\ \beta_{t+1} &= \beta_t + \eta_t, \end{aligned} \quad (2)$$

where $\eta_t \sim N(0, D)$, with D being $p \times p$ diagonal matrix with diagonal elements $\sigma_{i,\eta}^2$, $i = 1, \dots, p$, and define a prior distribution for the first time point β_1 as $N(\mu_{\beta_1}, \sigma_{\beta_1}^2)$. The bottom equation in

E-mail address: jouni.helske@jyu.fi.

(2) defines a random walk process for the regression coefficients with D defining the degree of variability of the coefficients (with $D = 0$ the model collapses to basic regression model).

Our goal is a Bayesian estimation of the unknown regression coefficients β_1, \dots, β_T and standard deviations $\sigma = (\sigma_\epsilon, \sigma_{1,\eta}, \dots, \sigma_{p,\eta})$. Although in principle we can estimate these using the general-purpose Markov chain Monte Carlo (MCMC) software such as Stan [5] or BUGS [6], these standard implementations can be computationally inefficient and prone to severe problems related to the convergence of the underlying MCMC algorithm due to the nature of our models of interest. For example in (block) Gibbs sampling approach we target the joint posterior $p(\beta, \sigma | y)$ by sampling from $p(\beta | \sigma, y)$ and $p(\sigma | \beta, y)$. But because of strong autocorrelations between the coefficients β at different time points, as well as with the associated standard deviation parameters, this MCMC scheme can lead to slow mixing. Also, the total number of parameters to sample increases with the number of data points T . Although the Hamiltonian Monte Carlo algorithms offered by Stan are typically more efficient exploring high-dimensional posteriors than Gibbs-type algorithms, we still encounter similar problems.

An alternative solution used by the R [7] package walker is based on the property that model (2) can be written as a linear gaussian state space model (see, e.g., Section 3.6.1 in [8]). This allows us to marginalize the regression coefficients β during the MCMC sampling by using the Kalman filter leading to a fast and accurate inference of marginal posterior $p(\sigma | y)$. Then, the corresponding joint posterior $p(\sigma, \beta | y) = p(\beta | \sigma, y)p(\sigma | y)$ can be obtained by simulating the regression coefficients given marginal posterior of standard deviations. This sampling can be performed for example by simulation smoothing algorithm [9].

The marginalization of regression coefficients cannot be directly extended to generalized linear models such as Poisson regression, as the marginal log-likelihood is intractable. However, it is possible to use Gaussian approximation of this exponential state space model [10], and the resulting samples from the approximating posterior can then be weighted using the importance sampling type correction [11], leading again to asymptotically exact inference.

When modelling the regression coefficients as a simple random walk, the posterior estimates of these coefficients can have large short-term variation which might not be realistic in practice. One way of imposing more smoothness for the estimates is to switch from random walk coefficients to integrated second order random walk coefficients, defined as

$$\beta_{t+1} = \beta_t + \nu_t,$$

$$\nu_{t+1} = \nu_t + \xi_t,$$

with $\xi_t \sim N(0, D_\xi)$. This is a local linear trend model [12] (also known as an integrated random walk), with the restriction that there is no noise on the β level. For this model, we can apply the same estimation techniques as for the random walk coefficient case. More complex patterns for β are also possible. For example, we can define that $\beta_{t+1} \sim N(\beta_t, \gamma_t \sigma_\eta)$ where $\gamma_1, \dots, \gamma_T$ is known monotonically decreasing sequence of values leading to a case where β_t gradually converges to constant over time.

2. Software description

A stable version of walker is available at CRAN,¹ while the current development version can be installed from Github.² The MCMC sampling is handled by rstan, the R interface to Stan,

while the model definitions and analysis of the results are performed in R, leading to fast and flexible modelling.

For defining the models, the walker package uses similar Wilkinson–Rogers model formulation syntax [13] as, for example, basic linear model function lm in R, but the formula for walker also recognizes two custom functions, rw1 and rw2 for random walk and integrated random walk respectively. For example, by typing

```
fit <- walker(y ~ 0 + x +
  rw1(~ z, beta = c(0, 1), sigma = c(2, 1)),
  sigma_y = c(2, 1))
```

we define a model with time-invariant coefficient for predictor x , and first order random walk coefficients for z and time-varying intercept term. Priors for β (including the intercept) and σ are defined as vectors of length two which define the mean and standard deviation of the normal distribution and shape and rate of the Gamma distribution respectively.

Function walker creates the model based on the formula and the prior definitions, and then calls the sampling function from the rstan package. The resulting posterior samples can be then converted to a data frame format using the function as.data.frame for easy visualization and further analysis.

In addition to the main functions walker for the Gaussian case and walker_glm for the Poisson and negative binomial models, the package contains additional functions for visualization of the results (e.g., plot_coef and pp_check) and out-of-sample prediction (function predict). Also, as the modelling functions return the full stanfit used in MCMC sampling, this object can be analysed using many general diagnostic and graphical tools provided by several Stan related R packages such as ShinyStan [14].

3. Illustrative examples

As an illustrative example, let us consider a observations y of length $n = 100$, generated by random walk (i.e. time varying intercept) and two predictors. First we simulate the coefficients, predictors and the observations:

```
set.seed(1)
n <- 100
beta1 <- cumsum(c(0.5, rnorm(n - 1, 0, sd = 0.05)))
beta2 <- cumsum(c(-1, rnorm(n - 1, 0, sd = 0.15)))
x1 <- rnorm(n, mean = 2)
x2 <- cos(1:n)
rw <- cumsum(rnorm(n, 0, 0.5))
signal <- rw + beta1 * x1 + beta2 * x2
y <- rnorm(n, signal, 0.5)
```

Then we can call function walker. As noted in Section 2 the model is defined as a simple formula object, and in addition to the prior definitions we can pass various arguments to the sampling method of rstan, such as the number of iterations iter and the number of chains chains to be used for the MCMC (default values for these are 2000 and 4 respectively).

```
fit <- walker(y ~ 0 + rw1(~ x1 + x2,
  beta = c(0, 10), sigma = c(2, 4)),
  sigma_y = c(2, 4), chains = 2, seed = 1)
```

The output of walker is walker_fit object, which is essentially a list with stanfit from Stan's sampling function, the original observations y and the covariate matrix xreg. This allows us to use all the postprocessing functions for stanfit objects.

¹ <https://cran.r-project.org/package=walker>.

² <https://github.com/helske/walker>.

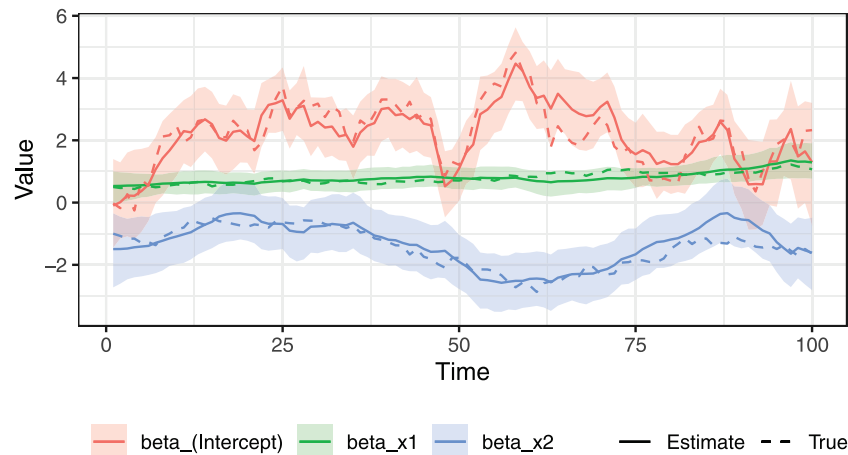


Fig. 1. Posterior means (solid lines) and 95% posterior intervals (shaded areas) of the time-varying regression coefficients and the true data generating values (dashed lines) of the illustrative example.

Fig. 1 shows how *walker* recovers the true coefficient processes relatively well: The posterior intervals contain the true coefficients and the posterior mean estimates closely follow the true values. For drawing the figure, we first use function `as.data.frame` to extract our posterior samples of the coefficients as data frame, then use packages `dplyr` [15] and `ggplot2` [16] to summarize and plot the posterior means and 95% posterior intervals respectively:

```
library(dplyr)
library(ggplot2)
sumr <- as.data.frame(fit, type = "tv") %>%
  group_by(variable, time) %>%
  summarise(mean = mean(value), lwr = quantile(value, 0.025),
            upr = quantile(value, 0.975))
sumr$true <- c(rw, beta1, beta2)

ggplot(sumr, aes(y = mean, x = time, colour = variable)) +
  geom_ribbon(aes(ymin = lwr, ymax = upr, fill = variable),
            colour = NA, alpha = 0.2) +
  geom_line(aes(linetype = "Estimate"), lwd = 1) +
  geom_line(aes(y = true, linetype = "True"), lwd = 1) +
  scale_linetype_manual(values = c("solid", "dashed")) +
  theme_bw() + xlab("Time") + ylab("Value") +
  theme(legend.position = "bottom",
        legend.title = element_blank())
```

More examples can be found in the package vignette and function documentation pages (e.g. typing `vignette("walker")` or `?walker_glm` in R), including a comparison between the marginalization approach of *walker* and “naive” implementation with Stan, and an example on the scalability.

4. Impact and conclusions

The *walker* package extends standard Bayesian generalized linear models to flexible time-varying coefficients case in a computationally efficient manner, which allows researchers in economics, social sciences and other fields to relax the sometimes unreasonable assumption of a stable, time-invariant relationship between the response variable and (some of) the predictors. Similar methods have been previously used in maximum likelihood setting for example in studying the diminishing effects of advertising [17] and demand for international reserves [18].

There are several ways how *walker* can be extended in the future. There are already some plans for additional forms of time-varying coefficients (such as a stationary autoregressive process), support for more priors and additional distributions for the response variables (e.g., negative binomial).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been supported by the Academy of Finland research grants 284513, 312605, 311877, and 331817.

References

- [1] Moryson M. *Testing for random walk coefficients in regression and state space models*. Springer Science & Business Media; 2012.
- [2] Robinson PM. Nonparametric estimation of time-varying parameters. In: *Statistical analysis and forecasting of economic structural change*. Springer; 1989, p. 253–64.
- [3] Dagum P, Galper A, Horvitz E. Dynamic network models for forecasting. In: *Proceedings of the eighth conference on uncertainty in artificial intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1992, p. 41–8.
- [4] Harvey AC, Phillips GDA. The estimation of regression models with time-varying parameters. In: Deistler M, Fürst E, Schwödiauer G, editors. *Games, economic dynamics, and time series analysis*. Heidelberg: Physica-Verlag HD; 1982, p. 306–21.
- [5] Stan Development Team. The Stan C++ library. 2016, version 2.15.0. URL <http://mc-stan.org/>.
- [6] Lunn DJ, Thomas A, Best N, Spiegelhalter D. WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility. 2000, <http://dx.doi.org/10.1023/A:1008929526011>.
- [7] R Core Team. R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing; 2020, URL <https://www.R-project.org/>.
- [8] Durbin J, Koopman SJ. *Time series analysis by state space methods*. 2nd ed. New York: Oxford University Press; 2012.
- [9] Durbin J, Koopman SJ. A simple and efficient simulation smoother for state space time series analysis. *Biometrika* 2002;89:603–15.
- [10] Durbin J, Koopman SJ. Monte Carlo maximum likelihood estimation for non-Gaussian state space models. *Biometrika* 1997;84(3):669–84.
- [11] Vihola M, Helske J, Franks J. Importance sampling type estimators based on approximate marginal Markov chain Monte Carlo. *Scandinavian J. Stat.* 2020;47(4):1339–76. <http://dx.doi.org/10.1111/sjos.12492>.
- [12] Harvey AC. *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press; 1990.
- [13] Wilkinson GN, Rogers CE. Symbolic description of factorial models for analysis of variance. *J R Stat Soc Ser C Appl Stat* 1973;22(3):392–9, URL <http://www.jstor.org/stable/2346786>.
- [14] Stan Development Team. *shinystan: Interactive visual and numerical diagnostics and posterior analysis for Bayesian models*. 2017, R package version 2.3.0. URL <http://mc-stan.org/>.

- [15] Wickham H, François R, Henry L, Müller K. dplyr: A grammar of data manipulation. 2018, R package version 0.7.6. URL <https://CRAN.R-project.org/package=dplyr>.
- [16] Wickham H. Ggplot2: Elegant graphics for data analysis. Springer-Verlag New York; 2016, URL <https://ggplot2.tidyverse.org>.
- [17] Kinnucan HW, Venkateswaran M. Generic advertising; and the structural heterogeneity hypothesis. Can J Agric Econ Rev Can Agroeconomie 1994;42(3):381–96. <http://dx.doi.org/10.1111/j.1744-7976.1994.tb00032.x>.
- [18] Bahmani-Oskooee M, Brown F. Kalman filter approach to estimate the demand for international reserves. Appl Econ 2004;36(15):1655–68. <http://dx.doi.org/10.1080/0003684042000218543>.