# BayesHMM: Full Bayesian Inference for Hidden Markov Models

Luis Damiano [1] Michael Weylandt [2] Brian Peterson [3]

[1]Iowa State University (Statistics) [2]Rice University (Statistics) [3]University of Washington (Computational Finance & Risk Management)

## Hidden Markov Models [5]

Observation model: $p(\mathbf{y}_t|z_t)$, where $\mathbf{y}_t$ are the observations, emissions or output.

Homogeneous state model: discrete-time, discrete-state first-order Markov chain $z_t \in \{1, \dots, K\}$ driven by $p(z_t|z_{t-1})$, where $K$ is the number of latent states.

Heterogeneous state model: alternatively, time-varying covariates $\mathbf{x}_t \in \mathbb{R}^M$ may be used to drive the transition $p(z_t|\mathbf{u}_t, z_{t-1} = i) = \text{softmax}(\mathbf{u}_t\beta_i^u)$ [3].

Joint posterior density:

$$p(\mathbf{z}_{1:T}, \mathbf{y}_{1:T}) = p(\mathbf{z}_{1:T})p(\mathbf{y}_{1:T}|\mathbf{z}_{1:T}) = \left[ p(z_1) \prod_{t=2}^{T} p(z_t|z_{t-1}) \right] \left[ \prod_{t=1}^{T} p(\mathbf{y}_t|z_t) \right].$$

## Generative models

1. Generate parameters according to the priors $\theta^{(0)} \sim p(\theta)$. 2. Generate the hidden path $\mathbf{z}_{1:T}^{(0)}$ according to the transition model parameters. 3. Generate the observed quantities based on the sampling distribution $\mathbf{y}_t^{(0)} \sim p(\mathbf{y}_t|\mathbf{z}_{1:T}^{(0)}, \theta^{(0)})$.

## Inference & Learning

Several quantities of interest can be inferred via different algorithms. Our software contains the implementation of the most relevant methods for unsupervised data: forward [1], forward-backward [1, 2] and Viterbi decoding algorithms [6].

Table 1. Hidden quantities & inference algorithm. Time complexity is $O(K^2T)$ [4].

| Name | Hidden Quantity | Availability | Algorithm |
|---|---|---|---|
| Filtering | $p(z_t\|\mathbf{y}_{1:t})$ | $t$ (online) | Forward |
| Smoothing | $p(z_t\|\mathbf{y}_{1:T})$ | $T$ (offline) | Forward-backward |
| Fixed lag smoothing | $p(z_{t-\ell}\|\mathbf{y}_{1:t})$, $\ell \geq 1$ | $t + \ell$ (lag'd) | Forward-backward |
| State prediction | $p(z_{t+h}\|\mathbf{y}_{1:t})$, $h \geq 1$ | $t$ (online) | Forward-propagation |
| Observation prediction | $p(y_{t+h}\|\mathbf{y}_{1:t})$, $h \geq 1$ | $t$ (online) | Forward-propagation |
| MAP Estimation | $\text{argmax}_{\mathbf{z}_{1:T}}\, p(\mathbf{z}_{1:T}\|\mathbf{y}_{1:T})$ | $T$ (offline) | Viterbi decoding |
| Marginal evidence | $p(\mathbf{y}_{1:T})$ | $t$ (online) | Forward |

## References

[1] Leonard E. Baum and J. A. Eagon.
An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology.
*Bulletin of the American Mathematical Society*, 73(3):360–364, may 1967.

[2] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss.
A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains.
*The Annals of Mathematical Statistics*, 41(1):164–171, feb 1970.

[3] Yoshua Bengio and Paolo Frasconi.
An input output hmm architecture.
In *Proceedings of the 7th International Conference on Neural Information Processing Systems (NIPS 1994)*, pages 427–434, 1994.

[4] Kevin P. Murphy.
*Machine Learning: A Probabilistic Perspective.*
MIT Press Ltd, 2012.

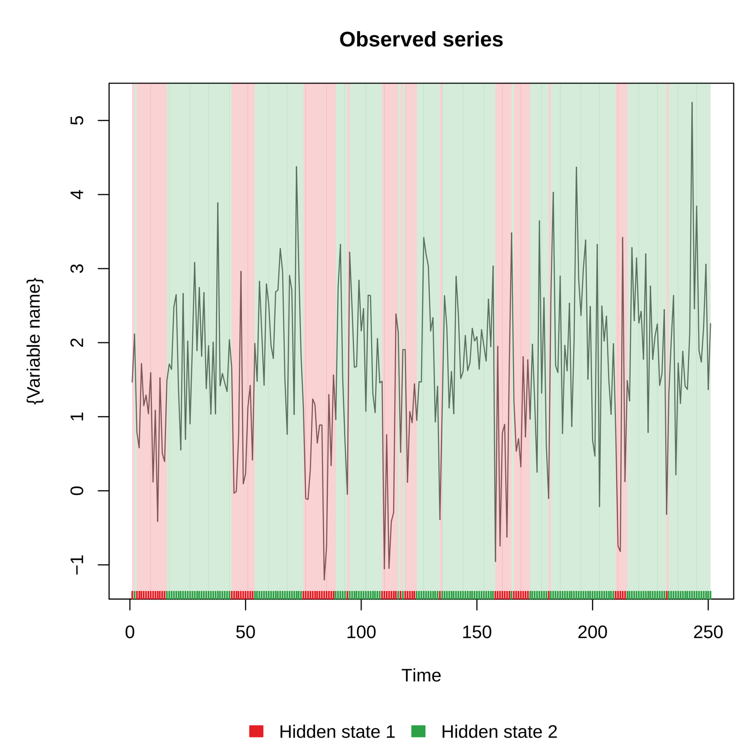
[5] Lawrence R. Rabiner.
A tutorial on hidden markov models and selected applications in speech recognition.
In *Readings in Speech Recognition*, pages 267–296. Elsevier, 1990.

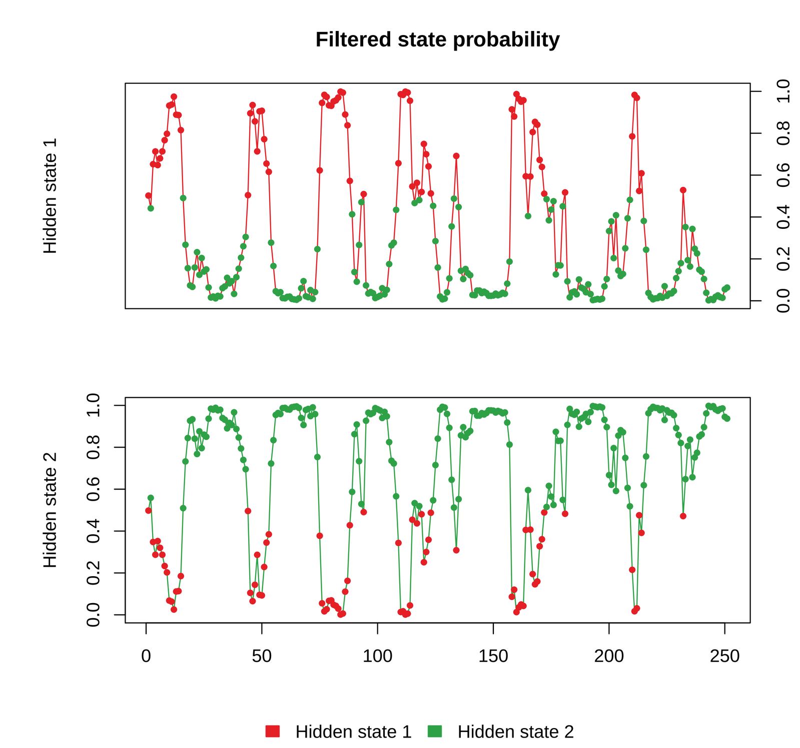
[6] A. Viterbi.
Error bounds for convolutional codes and an asymptotically optimum decoding algorithm.
*IEEE Transactions on Information Theory*, 13(2):260–269, apr 1967.

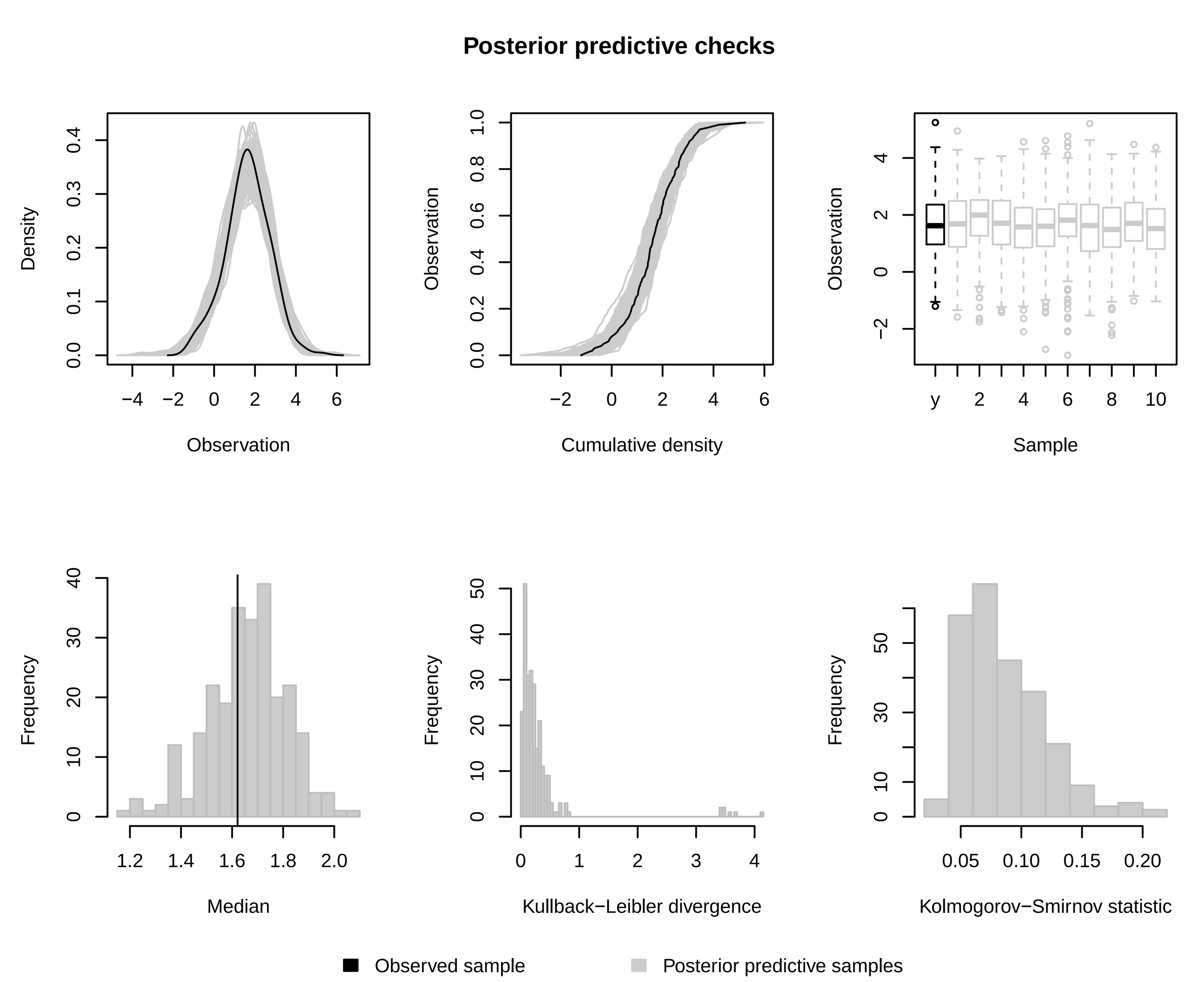
## Sample use

```
devtools::install_github("luisdamiano/BayesHMM") # Soon on CRAN
library(BayesHMM)

mySpec <- hmm(
  K = 2, R = 1,
  observation = Gaussian(
    mu    = Gaussian(mu = 0, sigma = 10, ordered = TRUE),
    sigma = Cauchy(mu = 0, sigma = 10, bounds = list(0, NULL))
  ),
  initial     = Dirichlet(alpha = c(1, 1)),
  transition  = Dirichlet(alpha = c(1, 1)),
  name = "Univariate Gaussian Dummy Model"
)

myModel <- compile(mySpec)
myFit   <- draw_samples(
  myModel, y = ySim, # ySim is numeric vector
  seed = 9000, iter = 500, chains = 1
)
```

## Sample visualizations




## Sample printout

One-stop print function: ☑ model description, ☑ Monte Carlo posterior estimates, ☑ elapsed time, ☑ MCMC convergence diagnostics, ☑ reproducibility notes.

```
----------------------------------------------------------------------
UNIVARIATE GAUSSIAN DUMMY MODEL
----------------------------------------------------------------------
Univariate observations (R = 1).
Observation model for Variable 1 in State 1
Variable Density: Gaussian (-infty, infty)
  Free parameters: 2 (mu, sigma)
    mu    : real mu11;
            Prior Density: Gaussian (-infty, infty)
                Fixed parameters: 2 (mu = 0, sigma = 10),
    sigma : real<lower = 0> sigma11;
            Prior Density: Cauchy [0, infty)
                Fixed parameters: 2 (mu = 0, sigma = 10)

Observation model for Variable 1 in State 2
Variable Density: Gaussian (-infty, infty)
  Free parameters: 2 (mu, sigma)
    mu    : real<lower = mu11> mu21;
            Prior Density: Gaussian [mu11, infty)
                Fixed parameters: 2 (mu = 0, sigma = 10),
    sigma : real<lower = 0> sigma21;
            Prior Density: Cauchy [0, infty)
                Fixed parameters: 2 (mu = 0, sigma = 10)

         P Mean  MCSE  P SD PI2.5 PI25.0 PI50.0 PI75.0 PI97.5    ESS  Rhat
mu11      0.806 0.036 0.294 0.135  0.655  0.820  0.988  1.284 66.538 1.017
mu21      1.993 0.013 0.130 1.783  1.895  1.972  2.086  2.243 99.725 1.002
sigma11   1.037 0.017 0.146 0.767  0.944  1.008  1.097  1.407 76.824 0.993
sigma21   0.910 0.009 0.081 0.730  0.872  0.920  0.956  1.035 80.254 0.994
----------------------------------------------------------------------
Initial distribution model
Prior Density: Dirichlet (-infty, infty)
    Fixed parameters: 1 (alpha = [1, 1])

         P Mean  MCSE  P SD PI2.5 PI25.0 PI50.0 PI75.0 PI97.5    ESS  Rhat
pi[1]     0.478 0.033 0.325 0.002  0.212  0.430  0.791  0.981 98.751 0.994
pi[2]     0.522 0.033 0.325 0.019  0.209  0.570  0.788  0.998 98.751 0.994
----------------------------------------------------------------------
Transition model
Prior Density: Dirichlet (-infty, infty)
    Fixed parameters: 1 (alpha = [1, 1])

         P Mean  MCSE  P SD PI2.5 PI25.0 PI50.0 PI75.0 PI97.5    ESS  Rhat
A[1,1]    0.805 0.009 0.088 0.594  0.762  0.827  0.874  0.928 87.863 1.010
A[2,1]    0.090 0.007 0.063 0.026  0.051  0.075  0.115  0.225 89.778 0.992
A[1,2]    0.195 0.009 0.088 0.072  0.126  0.173  0.238  0.406 87.863 1.010
A[2,2]    0.910 0.007 0.063 0.775  0.885  0.925  0.949  0.974 89.778 0.992
----------------------------------------------------------------------
Model ran on 2019-04-16 at 08:50:36 using seed 9000.
Sample size: 250 kept iterations (500 total, 250 warmup, thin every 1).
Time elapsed in minutes:
- Chain 1: warmup 1.126, sample 0.555.
----------------------------------------------------------------------
Convergence diagnostics
Chain 1: 79 iterations hitting max number of leapfrogs.
----------------------------------------------------------------------
Notes for reproducibility:
 - Printed on 2019-04-16 at 08:50:36.
 - Ubuntu 18.04.2 LTS x86_64-pc-linux-gnu (64-bit)
 - R version 3.5.3 (2019-03-11)
 - BayesHMM v0.0.1 (Build R 3.5.3; ; 2019-04-16 13:48:47 UTC; unix)
 - rstan v2.18.2 (Build R 3.5.3; x86_64-pc-linux-gnu; 2019-04-08 00:19:31 UTC)
 - BayesHMM 0.0.1, Rcpp 1.0.0
```

## Choice of densities

Observation model: univariate (Bernoulli, Beta, Binomial, Categorical, Cauchy, Dirichlet, Gaussian, Multinomial, Negative Binomial, Poisson, Student), multivariate (Gaussian, Student), Regressions (Bernoulli, Binomial, Softmax, Gaussian), prior-only (LKJ, Wishart). Transition model: Dirichlet, Softmax regression.