Package ‘npfseir’

May 9, 2026

Title Nested Particle Filter for Stochastic SEIR Epidemic Models

Version 0.2.1

Date 2026-04-22

Description Implements the online Bayesian inference framework for joint state and parameter estimation in a stochastic Susceptible-Exposed-Infectious-Recovered (SEIR) epidemic model with a time-varying transmission rate. The log-transmission rate is modelled as a latent Ornstein-Uhlenbeck (OU) process with exact Gaussian discrete-time transitions. Inference is performed via the nested particle filter (NPF) of Crisan and Miguez (2018) <doi:10.3150/17-BEJ954>, which maintains an outer particle layer over the OU hyperparameters and, for each outer particle, an inner bootstrap filter over epidemic states. The Cori-style renewal-equation estimator follows Cori et al. (2013) <doi:10.1093/aje/kwt133>. The package also provides utilities for simulation, posterior summarisation, and forecasting.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats, graphics, grDevices

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Weinan Wang [aut, cre]

Maintainer Weinan Wang <ww@ou.edu>

Repository CRAN

Date/Publication 2026-04-24 18:40:07 UTC

Contents

`npfseir-package` *npfseir: Nested Particle Filter for Stochastic SEIR Epidemic Models*

Description

Implements the online Bayesian inference framework for stochastic SEIR epidemic models with latent Ornstein-Uhlenbeck transmission dynamics, as described in Feng and Wang (2025).

Main functions

`npf_seir` Run the nested particle filter.

`seir_simulate` Simulate a stochastic SEIR trajectory.

`cori_rt` Cori-style renewal R_t estimator (illustrative).

`ou_params` Exact OU discrete-time transition parameters.

S3 methods for `npf_seir` objects

`print`, `summary`, `plot`, `predict`.

Author(s)

Maintainer: Weinan Wang <ww@ou.edu>

References

Feng, W. and Wang, W. (2025). "Bayesian sequential inference for a stochastic SEIR model with latent transmission dynamics." *Preprint*.

Crisan, D. and Miguez, J. (2018). "Nested particle filters for online parameter estimation in discrete-time state-space Markov models." *Bernoulli*, 24(4B):3039–3086. doi:10.3150/17BEJ954

cori_rt *Cori-style renewal-equation R_t estimator*

Description

An in-house implementation of the sliding-window renewal-equation R_t estimator following Cori et al. (2013).

Usage

```
cori_rt(
  incidence,
  tau = 7L,
  si_mean = 5.5,
  si_sd = 2.5,
  prior_a = 1,
  prior_b = 5
)
```

Arguments

`incidence`	Numeric vector of length T. Daily case counts.
`tau`	Integer. Sliding window length (days). Default 7.
`si_mean`	Positive numeric. Serial interval mean (days). Default 5.5.
`si_sd`	Positive numeric. Serial interval SD (days). Default 2.5.
`prior_a`	Positive numeric. Gamma prior shape on R_t. Default 1.
`prior_b`	Positive numeric. Gamma prior scale on R_t. Default 5.

Details

Important: This is *not* a call to the R **EpiEstim** package and numerical agreement with that package has not been independently verified. It is provided for *illustrative comparison of estimands* only: the renewal-equation R_t and the compartmental R_t target different quantities once susceptible depletion is non-negligible (see Section 6.2 of Feng and Wang 2025).

The estimator uses gamma-conjugate updating over a τ-day sliding window. The posterior is

$$R_t \mid \text{data} \sim \text{Gamma}(a + \sum I,\ [1/b + \Lambda]^{-1})$$

where $\Lambda = \sum_{s=t-\tau+1}^{t} \Lambda_s$ and $\Lambda_s = \sum_{k\geq 1} w_k I_{s-k}$ is daily infectiousness.

Value

A data frame with columns `t`, `Rt_mean`, `Rt_lo`, `Rt_hi` (NaN where incidence is too low to estimate).

References

Cori, A., Ferguson, N. M., Fraser, C. and Cauchemez, S. (2013). A new framework and software to estimate time-varying reproduction numbers during epidemics. *American Journal of Epidemiology*, 178(9):1505–1512.

Examples

```
set.seed(1)
inc <- rpois(60, lambda = c(rep(50, 20), seq(50, 200, length=20),
                             seq(200, 30, length=20)))
rt  <- cori_rt(inc, tau = 7, si_mean = 5.5, si_sd = 2.5)
plot(rt$t, rt$Rt_mean, type = "l", ylim = c(0, 4),
     xlab = "Day", ylab = expression(R[t]),
     main = "Cori-style Rt estimate")
abline(h = 1, lty = 2)
```

`npf_seir` *Run the Nested Particle Filter for a stochastic SEIR model*

Description

Performs online Bayesian joint state and parameter estimation for the stochastic SEIR model with latent Ornstein-Uhlenbeck transmission dynamics, using the nested particle filter (NPF) of Crisan and Miguez (2018).

Usage

```
npf_seir(
  observations,
  fixed,
  K = 100L,
  M = 200L,
  tau_outer = 0.3,
  x0 = NULL,
  seed = NULL,
  h_min = 0.01,
  h_max = 0.1,
  rho = 0.995,
  verbose = FALSE
)
```

Arguments

`observations`	Numeric vector of length T. Daily confirmed case counts $P_1, \ldots, P_T$.
`fixed`	Named list of fixed model parameters (see `seir_simulate`).
`K`	Integer. Number of outer (parameter) particles. Default 100.
`M`	Integer. Number of inner (state) particles per outer particle. Default 200.

`tau_outer` Numeric in (0,1). ESS threshold for outer resampling. Outer resampling is triggered when $\mathrm{ESS}_{\mathrm{outer}} < \tau_{\mathrm{outer}} \cdot K$. Default 0.3.

`x0` Numeric vector of length 5, or `NULL`. Initial state $(S_0, E_0, I_0, R_0, \Psi_0)$. If `NULL`, derived from `observations[1]` via quasi-steady-state initialization; in that case the first observation is used only for initialization and excluded from the filtering updates.

`seed` Integer or `NULL`. Random seed for reproducibility.

`h_min, h_max, rho` Jittering bandwidth schedule parameters. The bandwidth at step n is $h_n = h_{min} + (h_{max} - h_{min})\rho^n$. Defaults: `h_min = 0.01`, `h_max = 0.10`, `rho = 0.995`.

`verbose` Logical. Print progress every 50 steps. Default `FALSE`.

Value

An object of class `"npf_seir"`, a list with components:

`Rt_mean`, `Rt_lo`, `Rt_hi` Numeric vectors of length T. Posterior mean and 95\ $R_t = \beta_t S_t/((\gamma + \delta)N)$.

`beta_mean`, `beta_lo`, `beta_hi` Posterior summaries for β_t.

`theta_mean` Matrix of size $T \times 3$. Weighted posterior mean of $\theta = (\kappa, \sigma_\Psi, \mu_\Psi)$ at each step.

`final_theta` Matrix $K \times 3$. Outer particles at time T.

`final_w` Numeric vector of length K. Outer weights at time T.

`final_inner` List of length K. Final inner state particles used for posterior predictive simulation.

`outer_ess` Numeric vector of length T. Outer ESS trace.

`observations`, `fixed`, `call` Input metadata.

References

Crisan, D. and Miguez, J. (2018). Nested particle filters for online parameter estimation in discrete-time state-space Markov models. *Bernoulli*, 24(4A):3039–3086.

Feng, W. and Wang, W. (2025). Bayesian sequential inference for a stochastic SEIR model with latent transmission dynamics. *Preprint*.

Examples

```
# Simulate and recover
fixed <- list(eps = 1/5, gamma = 1/10, delta = 1/(70*365),
              b = 1/(70*365), q = 0.3, N = 1e6, sigma_comp = 0.01)
x0    <- c(1e6 - 100, 50, 50, 0, log(0.25))
set.seed(1)
traj  <- seir_simulate(200, kappa=0.05, sigma_psi=0.10,
                       mu_psi=log(0.25), x0=x0, fixed=fixed)
fit   <- npf_seir(traj$obs[-1], fixed=fixed, K=50, M=100, seed=42)
plot(fit)
summary(fit)
```

ou_params *Compute exact Ornstein-Uhlenbeck discrete-time transition parameters*

Description

For the continuous-time OU SDE $d\Psi_t = \kappa(\mu_\Psi - \Psi_t)\,dt + \sigma_\Psi\,dW_t$, the exact one-day transition is $\Psi_n|\Psi_{n-1} \sim N(\mu_\Psi + \alpha(\Psi_{n-1} - \mu_\Psi), \tau^2)$ with $\alpha = e^{-\kappa}$ and $\tau^2 = \sigma_\Psi^2(1 - e^{-2\kappa})/(2\kappa)$.

Usage

```
ou_params(kappa, sigma_psi)
```

Arguments

kappa	Positive mean-reversion speed.
sigma_psi	Positive continuous-time volatility.

Value

A named list with elements `alpha` and `tau`.

Examples

```
ou_params(kappa = 0.05, sigma_psi = 0.10)
```

plot.npf_seir *Plot method for npf_seir*

Description

Produces posterior summary plots for R_t, β_t, or both.

Usage

```
## S3 method for class 'npf_seir'
plot(x, burn = 0L, which = "Rt", dates = NULL, ...)
```

Arguments

x	An `npf_seir` object.
burn	Integer. Number of initial steps to exclude from the plot.
which	Character. "Rt" (default), "beta", or "both" to show both trajectories in a two-panel display.
dates	Optional Date vector with one entry per observation to use as the x-axis.
...	Additional arguments passed to `plot()`.

Value

Returns x invisibly. Called primarily for its side effect of producing a base-graphics plot. When which = "Rt" or "beta", a single panel is drawn showing the posterior mean trajectory and 95% credible band. When which = "both", a two-panel figure is produced showing R_t (top) and β_t (bottom).

---

predict.npf_seir — *Posterior predictive forecasts from an npf_seir object*

Description

Generates forward simulations from the filter's final filtered particles to produce a posterior predictive distribution over future case counts.

Usage

```
## S3 method for class 'npf_seir'
predict(object, horizon = 14L, n_draws = 1000L, ...)
```

Arguments

object	An npf_seir object.
horizon	Integer. Number of days ahead to forecast. Default 14.
n_draws	Integer. Number of Monte Carlo draws. Default 1000.
...	Ignored.

Value

A named list with components: mean (numeric vector of length horizon, posterior predictive mean), lo and hi (95\ (n_draws-by-horizon matrix of Monte Carlo draws).

Examples

```
fixed <- list(eps = 1/5, gamma = 1/10, delta = 1/(70*365),
              b = 1/(70*365), q = 0.3, N = 1e6, sigma_comp = 0.01)
x0    <- c(1e6 - 100, 50, 50, 0, log(0.25))
set.seed(1)
traj  <- seir_simulate(200, 0.05, 0.10, log(0.25), x0, fixed)
fit   <- npf_seir(traj$obs[-1], fixed, K=50, M=100, seed=42)
fc    <- predict(fit, horizon = 14)
print(fc$mean)
```

`print.npf_seir` *Print method for npf_seir*

Description

Print method for npf_seir

Usage

```
## S3 method for class 'npf_seir'
print(x, ...)
```

Arguments

`x`	An `npf_seir` object.
`...`	Ignored.

Value

Returns `x` invisibly. Called primarily for its side effect of printing a concise summary to the console, including the number of observations and particles, and the posterior means of the Ornstein-Uhlenbeck hyperparameters κ, σ_Ψ, and μ_Ψ at the final time step.

`seir_simulate` *Simulate a stochastic SEIR epidemic with latent OU transmission*

Description

Generates a synthetic epidemic trajectory from the stochastic SEIR model with exact Ornstein-Uhlenbeck log-transmission dynamics, as described in Section 2 of Feng and Wang (2025).

Usage

```
seir_simulate(n_steps, kappa, sigma_psi, mu_psi, x0, fixed)
```

Arguments

`n_steps`	Integer. Number of days to simulate.
`kappa`	Positive numeric. OU mean-reversion speed.
`sigma_psi`	Positive numeric. OU continuous-time volatility.
`mu_psi`	Numeric. OU long-run mean of log-transmission.
`x0`	Numeric vector of length 5: initial state $(S_0, E_0, I_0, R_0, \Psi_0)$.
`fixed`	Named list of fixed model parameters. Must contain:
	eps Incubation rate ϵ (1/mean incubation period).

gamma Recovery rate γ (1/mean infectious period).
delta Background mortality rate δ.
b Birth rate b (usually equal to δ).
q Case detection probability $q \in (0, 1]$.
N Reference population size N.
sigma_comp Compartment diffusion coefficient $\sigma_\cdot$.

Value

A data frame with `n_steps + 1` rows (day 0 to day `n_steps`) and columns: `day`, `S`, `E`, `I`, `R`, `Psi`, `beta` (e^{Ψ_t}), `obs` (simulated observed daily counts drawn from a Poisson distribution with mean $q\epsilon E_t$).

References

Feng, W. and Wang, W. (2025). Bayesian sequential inference for a stochastic SEIR model with latent transmission dynamics. *Preprint*.

Examples

```
fixed <- list(eps = 1/5, gamma = 1/10, delta = 1/(70*365),
              b = 1/(70*365), q = 0.3, N = 1e6, sigma_comp = 0.01)
x0 <- c(S = 1e6 - 100, E = 50, I = 50, R = 0, Psi = log(0.25))
traj <- seir_simulate(n_steps = 200, kappa = 0.05, sigma_psi = 0.10,
                      mu_psi = log(0.25), x0 = x0, fixed = fixed)
plot(traj$day, traj$obs, type = "l", xlab = "Day",
     ylab = "Daily cases", main = "Simulated epidemic")
```

`summary.npf_seir` *Summary method for npf_seir*

Description

Summarises posterior parameter estimates and the range of R_t.

Usage

```
## S3 method for class 'npf_seir'
summary(object, burn = 0L, ...)
```

Arguments

`object`	An `npf_seir` object.
`burn`	Integer. Number of initial steps to exclude from summaries. Default 0.
`...`	Ignored.

Value

A named list of class "npf_seir_summary" returned invisibly with components:

theta Named numeric vector of length 3 giving the weighted posterior mean of the OU hyperparameters $(\kappa, \sigma_\Psi, \mu_\Psi)$ at the final time step.

ou Named list with elements alpha and tau, the exact discrete-time OU transition parameters derived from theta; see ou_params.

The function is also called for its side effect of printing a formatted summary to the console, including posterior parameter estimates and the range of R_t over the retained time steps.

Index