Package ‘epkde’

September 10, 2026

Type Package

Title Bayesian Bandwidth Selection for Multivariate KDE via Expectation Propagation

Version 0.1.0

Author Maurizio Filippone [aut, cre]

Maintainer Maurizio Filippone <maurizio.filippone@kaust.edu.sa>

Description Implements the approximate Bayesian method for bandwidth selection in multivariate kernel density estimation (KDE) proposed in Filippone & Sanguinetti (2011) <doi:10.1016/j.csda.2011.05.023>. The method uses the Expectation Propagation (EP) algorithm to approximate the posterior distribution of the inverse bandwidth (precision matrix) under a leave-one-out cross-validated likelihood. Three covariance structures are supported: isotropic (scalar precision), diagonal, and full precision matrix. Online Bayesian updating is supported for the isotropic case. The approximate posterior can be used for bandwidth selection, model comparison (via the model evidence / Bayes factor), and online learning.

Depends R (>= 4.0.0)

License GPL-3

Encoding UTF-8

Suggests ks, testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

RoxygenNote 7.3.2

VignetteBuilder knitr

URL https://github.com/mauriziofilippone/epkde

BugReports https://github.com/mauriziofilippone/epkde/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2026-09-10 13:10:08 UTC

Contents

`ep_kde_diagonal` *EP inference for diagonal KDE bandwidth*

Description

Runs Expectation Propagation to approximate the posterior distribution of the diagonal kernel precision $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)$ in multivariate Gaussian KDE. Each dimension is assigned an independent Gamma prior.

Usage

```
ep_kde_diagonal(x, prior_shape, prior_rate, max_iter = 100, tol = 1e-06)
```

Arguments

`x`	Numeric matrix of training data ($n \times d$).
`prior_shape`	Numeric vector of length d; shape parameters a_{0k} for the Gamma prior on each λ_k. Use `rep(0, d)` for an improper flat prior.
`prior_rate`	Numeric vector of length d; rate parameters b_{0k} for the Gamma prior on each λ_k. Use `rep(0, d)` for an improper flat prior.
`max_iter`	Maximum number of EP outer iterations. Default 100.
`tol`	Convergence tolerance. Default `1e-6`.

Value

A named list with elements:

`post_shape` Length-d vector of posterior Gamma shapes.

`post_rate` Length-d vector of posterior Gamma rates.

`log_evidence` Approximate log model evidence.

`log_norm_const` Log normalization constant of the posterior.

`shape_factors` $n \times d$ matrix of EP factor shapes.

`rate_factors` $n \times d$ matrix of EP factor rates.

`log_norm_factors` Length-n vector of EP factor log normalization constants.

References

Filippone, M. & Sanguinetti, G. (2011). Approximate inference of the bandwidth in multivariate kernel density estimation. *Computational Statistics & Data Analysis*, 55(12), 3104-3122.

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 2), ncol = 2)
fit <- ep_kde_diagonal(x, prior_shape = rep(1, 2), prior_rate = rep(1, 2))
cat("Posterior mean precision:", fit$post_shape / fit$post_rate, "\n")
```

`ep_kde_full` *EP inference for full KDE bandwidth (precision matrix)*

Description

Runs Expectation Propagation to approximate the posterior distribution of the full $d \times d$ kernel precision matrix Λ in multivariate Gaussian KDE. The approximate posterior is a Wishart distribution $q(\Lambda) = \mathcal{W}(\Lambda \mid \Sigma^{-1}, \nu)$.

Usage

```
ep_kde_full(x, prior_cov = NULL, prior_nu = 0, max_iter = 100, tol = 1e-06)
```

Arguments

`x`	Numeric matrix of training data ($n \times d$).
`prior_cov`	$d \times d$ prior covariance matrix $\Sigma_0 = \Lambda_0^{-1}$. Pass `NULL` or a large-diagonal matrix for a diffuse prior. Ignored when `prior_nu = 0`.
`prior_nu`	Wishart degrees-of-freedom parameter ν_0 for the prior. Use 0 for an improper flat prior.
`max_iter`	Maximum number of EP outer iterations. Default 100.
`tol`	Convergence tolerance on ν. Default `1e-6`.

Value

A named list with elements:

`post_prec` Posterior Wishart precision parameter Λ (i.e., $E[\Lambda]/\nu$).

`post_nu` Posterior Wishart degrees-of-freedom ν.

`post_mean` Posterior mean precision matrix $E[\Lambda] = \nu\Lambda_{new}$.

`bandwidth_matrix` Posterior mean bandwidth matrix (inverse of `post_mean`).

`log_evidence` Approximate log model evidence.

`log_norm_const` Log normalization constant of the posterior.

Returns `NULL` with a warning if EP fails to converge.

References

Filippone, M. & Sanguinetti, G. (2011). Approximate inference of the bandwidth in multivariate kernel density estimation. *Computational Statistics & Data Analysis*, 55(12), 3104-3122.

Examples

```
set.seed(1)
x <- matrix(rnorm(50 * 2), ncol = 2)
fit <- ep_kde_full(x, prior_cov = NULL, prior_nu = 0)
cat("Posterior nu:", fit$post_nu, "\n")
print(fit$bandwidth_matrix)
```

---

ep_kde_isotropic — *EP inference for isotropic KDE bandwidth*

Description

Runs Expectation Propagation to approximate the posterior distribution of the scalar kernel precision λ (inverse bandwidth squared) in multivariate Gaussian KDE with an isotropic kernel. The approximate posterior is $\Gamma(\lambda \mid a, b)$ and the log model evidence is also returned.

Usage

```
ep_kde_isotropic(
  x,
  prior_shape = 0,
  prior_rate = 0,
  max_iter = 100,
  tol = 1e-09
)
```

Arguments

`x`	Numeric matrix of training data ($n \times d$) or a numeric vector for the univariate case.
`prior_shape`	Non-negative scalar; shape parameter a_0 of the Gamma prior on λ. Use 0 for an improper flat prior.
`prior_rate`	Non-negative scalar; rate parameter b_0 of the Gamma prior on λ. Use 0 for an improper flat prior.
`max_iter`	Maximum number of EP outer iterations. Default 100.
`tol`	Convergence tolerance on the change in posterior parameters. Default `1e-9`.

Value

A named list with elements:

`post_shape` Posterior Gamma shape a.

`post_rate` Posterior Gamma rate b.

`log_evidence` Approximate log model evidence.

`log_norm_const` Log normalization constant of the approximate posterior (for expert use).

`shape_factors` Length-n vector of EP factor shape parameters (one per data point).

`rate_factors` Length-n vector of EP factor rate parameters (one per data point).

`log_norm_factors` Length-n vector of EP factor log normalization constants (one per data point).

References

Filippone, M. & Sanguinetti, G. (2011). Approximate inference of the bandwidth in multivariate kernel density estimation. *Computational Statistics & Data Analysis*, 55(12), 3104-3122.

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 2), ncol = 2)   # 100 bivariate observations
fit <- ep_kde_isotropic(x, prior_shape = 1, prior_rate = 1)
cat("Posterior mean precision:", fit$post_shape / fit$post_rate, "\n")
cat("Log model evidence:", fit$log_evidence, "\n")
```

`ep_kde_online` *Online EP update for isotropic KDE bandwidth*

Description

Updates a previously computed EP approximation (from `ep_kde_isotropic`) to incorporate a new batch of m observations, without re-processing the original n data points from scratch.

Usage

```
ep_kde_online(
  x_old,
  x_new,
  prior_shape,
  prior_rate,
  fit_old,
  max_iter = 100,
  tol = 1e-09
)
```

Arguments

`x_old` Numeric matrix ($n \times d$) of the original training data.

`x_new` Numeric matrix ($m \times d$) of the new observations to incorporate.

`prior_shape`, `prior_rate`
Prior parameters (same values used in the original `ep_kde_isotropic` call).

`fit_old` The list returned by `ep_kde_isotropic` on `x_old`.

`max_iter`, `tol` Convergence control (see `ep_kde_isotropic`).

Value

A named list with the same structure as `ep_kde_isotropic`, but with `shape_factors`, `rate_factors`, and `log_norm_factors` of length $n + m$.

References

Filippone, M. & Sanguinetti, G. (2011). Approximate inference of the bandwidth in multivariate kernel density estimation. *Computational Statistics & Data Analysis*, 55(12), 3104-3122.

`kde_predict` *Evaluate kernel density estimate at test points*

Description

Given training data and a fitted precision parameter (from one of the `ep_kde_*` functions), evaluates the KDE

$$\hat{p}(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^{n} \mathcal{N}(\mathbf{x}; \mathbf{x}_j, \Lambda^{-1})$$

at a set of test points.

Usage

```
kde_predict(x_test, x_train, precision_matrix, log_scale = FALSE)
```

Arguments

`x_test` Numeric matrix ($m \times d$) of test locations, or a numeric vector for the univariate case.

`x_train` Numeric matrix ($n \times d$) of training data used to fit the KDE.

`precision_matrix`
$d \times d$ positive-definite precision matrix Λ. For the isotropic fit use `fit$post_shape / fit$post_rate * diag(d)`. For the diagonal fit use `diag(fit$post_shape / fit$post_rate)`. For the full fit use `fit$post_mean`.

`log_scale` Logical; if `TRUE` return log-densities. Default `FALSE`.

Value

Numeric vector of length m with the (log-)density values at `x_test`.

Examples

```
set.seed(1)
x_train <- matrix(rnorm(100 * 2), ncol = 2)
fit <- ep_kde_isotropic(x_train, prior_shape = 1, prior_rate = 1)
Lambda <- fit$post_shape / fit$post_rate * diag(2)
x_test <- matrix(c(0, 0, 1, 1), ncol = 2)
p_hat  <- kde_predict(x_test, x_train, Lambda)
```

`model_evidence` *Extract the log model evidence for model comparison*

Description

A convenience wrapper that returns the log model evidence from a fitted `ep_kde_*` object. The evidence can be used to compute Bayes factors comparing, e.g., isotropic vs. diagonal vs. full precision structures.

Usage

```
model_evidence(fit)
```

Arguments

`fit`	A list returned by `ep_kde_isotropic`, `ep_kde_diagonal`, or `ep_kde_full`.

Value

A single numeric value: the approximate log model evidence.

Examples

```
set.seed(1)
x <- matrix(rnorm(60 * 2), ncol = 2)
fit_iso  <- ep_kde_isotropic(x, prior_shape = 1, prior_rate = 1)
fit_diag <- ep_kde_diagonal(x,  prior_shape = rep(1, 2), prior_rate = rep(1, 2))
fit_full <- ep_kde_full(x, prior_cov = NULL, prior_nu = 0)

cat("Log evidence  isotropic:", model_evidence(fit_iso),  "\n")
cat("Log evidence  diagonal: ", model_evidence(fit_diag), "\n")
cat("Log evidence  full:     ", model_evidence(fit_full), "\n")

# Bayes factor: full vs. isotropic (on log scale)
cat("Log BF (full vs. iso):", model_evidence(fit_full) - model_evidence(fit_iso), "\n")
```

Index