Package 'truedist'

October 27, 2026

Title Estimating the Distribution of True Effects from Noisy Estimates

Version 0.1.0

Description Estimates the distribution of true effects (or any latent parameter) from estimates that each have a known, possibly different, standard error. The main function, em_smooth(), implements the smoothed EM (EMS) algorithm of Silverman, Jones, Wilson and Nychka (1990) <doi:10.1111/j.2517-6161.1990.tb01788.x> for a nonparametric mixing distribution on a grid. The smoothing uses one constant standard deviation, set from the typical standard error and the sample size.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 3.5.0)

Imports graphics, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Paul von Hippel [aut, cre]

Maintainer Paul von Hippel <ph3828@eid.utexas.edu>

Repository CRAN

Date/Publication 2026-10-27 18:30:02 UTC

Contents

density.normal_fit 2
em_smooth 2
normal_fit 4
plot.em_smooth 5

posterior_mean 6
summary.em_smooth 6
summary.normal_fit 7

Index **8**

`density.normal_fit` *Density of the fitted normal distribution of true effects*

Description

Density of the fitted normal distribution of true effects

Usage

```
## S3 method for class 'normal_fit'
density(x, at, ...)
```

Arguments

`x`	An object returned by `normal_fit()`.
`at`	Numeric vector of values at which to evaluate the density.
`...`	Unused.

Value

A numeric vector of densities, one per value of `at`.

`em_smooth` *Smoothed EM estimate of the distribution of true effects*

Description

Estimates the distribution G of latent true effects θ_i from estimates y_i that are normally distributed around θ_i with known standard errors s_i, which may differ from one estimate to the next. G is a nonparametric mixing distribution on a grid, estimated by the smoothed EM (EMS) algorithm of Silverman, Jones, Wilson and Nychka (1990): each EM update of the grid weights is followed by a Gaussian smoothing step.

Usage

```
em_smooth(estimate, se, smoothing = 0.1)
```

Arguments

`estimate`	Numeric vector of estimated effects.
`se`	Numeric vector of standard errors (positive, same length as `estimate`), treated as known.
`smoothing`	The constant c in the rule given in the Details section (at least 0.05). Larger values smooth more. With fewer than 3 estimates the function stops, and with fewer than 20 it warns that the result is very uncertain.

Details

The smoothing step uses a Gaussian kernel with one constant standard deviation, chosen from the data by a rule of thumb:

$$\mathrm{SD} = c \times \mathrm{median}(s_i) \times \sqrt{\log 500 / \log n}.$$

The SD grows with the typical standard error and, slowly, as the sample size n falls, following the logarithmic dependence on n that theory gives for deconvolution (Fan, 1991). The constant c is 0.1 by default, so that with $n = 500$ and a typical standard error of 0.15 the SD is about 0.015. In simulations with $n = 500$ and standard errors between 0.05 and 0.4, this recovered the true CDF as well as or better than 0.2, 0.33 or 0.5. This is a rule of thumb, not an optimal bandwidth: the best amount of smoothing depends on how smooth the true distribution is, which the data cannot reliably reveal. Features such as peak heights are sensitive to it, while the mean, SD, quantiles and CDF are much less so; consider rerunning with other values of `smoothing` as a sensitivity check.

Value

An object of class `"em_smooth"`, a list with components `grid`, `weights` (probability mass on each grid point), `density` (`weights` divided by the grid step), `loglik`, `iterations`, `converged`, `smoothing` (the value of `smoothing` used), `smoothing_sd` (the standard deviation of the smoothing kernel), and `data` (the input estimates and standard errors). Use `summary.em_smooth()` for the mean, SD, quantiles and exceedance probabilities.

References

Silverman, B. W., Jones, M. C., Wilson, J. D., and Nychka, D. W. (1990). A smoothed EM approach to indirect estimation problems, with particular reference to stereology and emission tomography. *Journal of the Royal Statistical Society, Series B*, 52(2), 271-303. doi:10.1111/j.2517-6161.1990.tb01788.x

Nychka, D. (1990). Some properties of adding a smoothing step to the EM algorithm. *Statistics & Probability Letters*, 9(2), 187-193.

Fan, J. (1991). On the optimal rates of convergence for nonparametric deconvolution problems. *Annals of Statistics*, 19(3), 1257-1272.

Examples

```
set.seed(1)
theta <- c(rnorm(300, 0, 0.1), rnorm(100, 0.8, 0.2))
se <- runif(400, 0.05, 0.4)
```

```
y <- rnorm(400, theta, se)
fit <- em_smooth(y, se)
fit
summary(fit, probs = c(0.1, 0.5, 0.9), thresholds = 0.2)
plot(fit)
```

`normal_fit` *Normal model for the distribution of true effects*

Description

Fits the model in which the true effects are normally distributed, $\theta_i \sim N(\mu, \tau^2)$, and each estimate is $y_i \sim N(\theta_i, s_i^2)$ with known standard error s_i. The mean μ and SD τ of the true effects are estimated by maximum likelihood, so that $y_i \sim N(\mu, \tau^2 + s_i^2)$. This is the parametric counterpart of `em_smooth()` and a useful baseline.

Usage

```
normal_fit(estimate, se)
```

Arguments

`estimate`	Numeric vector of estimated effects.
`se`	Numeric vector of standard errors (positive, same length as `estimate`), treated as known.

Value

An object of class `"normal_fit"`, a list with the estimated `mu` and `tau`, their standard errors `se_mu` and `se_tau` (from the observed information, treating estimates as independent), `loglik`, and `data`. Use `summary.normal_fit()` for the mean, SD, quantiles and exceedance probabilities, computed exactly from the normal distribution.

Examples

```
set.seed(1)
se <- runif(300, 0.05, 0.4)
y <- rnorm(300, rnorm(300, 0.2, 0.2), se)
fit <- normal_fit(y, se)
fit
density(fit, at = c(0, 0.2, 0.4))
```

`plot.em_smooth` *Plot an estimated true-effect distribution*

Description

Plot an estimated true-effect distribution

Usage

```
## S3 method for class 'em_smooth'
plot(
  x,
  rug = TRUE,
  xlab = "Effect",
  ylab = "Density",
  main = "Estimated distribution of true effects",
  ...
)
```

Arguments

`x`	An object returned by `em_smooth()`.
`rug`	Draw a rug of the observed estimates.
`xlab`, `ylab`, `main`	Axis labels and title.
`...`	Further arguments passed to `graphics::plot()`.

Value

`x`, invisibly.

Examples

```
set.seed(1)
se <- runif(200, 0.05, 0.4)
y <- rnorm(200, rnorm(200, 0.2, 0.2), se)
plot(em_smooth(y, se))
```

posterior_mean *Posterior mean of each true effect*

Description

Empirical Bayes posterior means of the true effects, using the estimated distribution from `em_smooth()` as the prior.

Usage

```
posterior_mean(object)
```

Arguments

`object`	An object returned by `em_smooth()`.

Value

A numeric vector with one posterior mean per estimate, in the order of the original data.

Examples

```
set.seed(1)
se <- runif(200, 0.05, 0.4)
y <- rnorm(200, rnorm(200, 0.2, 0.2), se)
fit <- em_smooth(y, se)
head(posterior_mean(fit))
```

summary.em_smooth *Summary of an estimated true-effect distribution*

Description

Summary of an estimated true-effect distribution

Usage

```
## S3 method for class 'em_smooth'
summary(object, probs = c(0.1, 0.25, 0.5, 0.75, 0.9), thresholds = 0, ...)
```

Arguments

`object`	An object returned by `em_smooth()`.
`probs`	Probabilities at which to report quantiles.
`thresholds`	Values `t` at which to report the probability that a true effect exceeds `t`.
`...`	Unused.

Value

A data frame with one row per statistic (mean, SD, quantiles, exceedance probabilities) and a column `estimate`.

Examples

```
set.seed(1)
se <- runif(200, 0.05, 0.4)
y <- rnorm(200, rnorm(200, 0.2, 0.2), se)
summary(em_smooth(y, se), probs = c(0.1, 0.5, 0.9), thresholds = 0.2)
```

`summary.normal_fit` *Summary of a fitted normal model*

Description

Summary of a fitted normal model

Usage

```
## S3 method for class 'normal_fit'
summary(object, probs = c(0.1, 0.25, 0.5, 0.75, 0.9), thresholds = 0, ...)
```

Arguments

`object`	An object returned by `normal_fit()`.
`probs`	Probabilities at which to report quantiles.
`thresholds`	Values `t` at which to report the probability that a true effect exceeds `t`.
`...`	Unused.

Value

A data frame with one row per statistic (mean, SD, quantiles, exceedance probabilities) and a column `estimate`, computed exactly from the fitted normal distribution. The format matches `summary.em_smooth()`.

Index

density.normal_fit, 2

em_smooth, 2
em_smooth(), *4–6*

graphics::plot(), *5*

normal_fit, 4
normal_fit(), *2*, *7*

plot.em_smooth, 5
posterior_mean, 6

summary.em_smooth, 6
summary.em_smooth(), *3*, *7*
summary.normal_fit, 7
summary.normal_fit(), *4*