Package ‘WSwavelet’

September 23, 2026

Type Package

Title Bayesian Wavelet Denoising with Wendland-Semicircle Slab Mixture

Version 0.1.0

Date 2026-09-12

Maintainer Nilotpal Sanyal <nsanyal@utep.edu>

Description Bayesian wavelet denoising using a resolution-adaptive spike-and-slab prior whose continuous slab is a mixture of a compactly supported Wendland-type density and the semicircle density. The package provides Gaussian and Laplace working-likelihood versions, empirical-Bayes fitting of resolution-specific slab weights, robust noise-scale estimation, posterior-mean coefficient shrinkage, reconstruction by the inverse discrete wavelet transform, and diagnostic tools.

License GPL-2 | GPL-3

URL https://nilotpalsanyal.github.io/WSwavelet/

BugReports https://github.com/nilotpalsanyal/WSwavelet/issues

Depends R (>= 4.1.0)

Imports wavethresh

Suggests testthat (>= 3.0.0), knitr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Author Nilotpal Sanyal [aut, cre]

Date/Publication 2026-09-23 02:40:09 UTC

Contents

`WSwavelet-package` *WSwavelet: Bayesian Wavelet Denoising with Wendland-Semicircle Slab Mixture*

Description

Bayesian wavelet denoising with a resolution-adaptive spike-and-slab prior. The continuous slab is a mixture of a compactly supported Wendland-type density and the semicircle density. The package provides Gaussian and Laplace working-likelihood versions, empirical-Bayes fitting of resolution-specific slab weights, robust noise-scale estimation, posterior-mean coefficient shrinkage, inverse-DWT reconstruction, and diagnostic tools.

Details

The primary user-facing function is `wswavelet`. It requires a finite numeric vector whose length is a dyadic integer, such as 128 or 1024. The function applies an orthogonal discrete wavelet transform using **wavethresh**, retains the scaling coefficients, shrinks detail coefficients, and reconstructs the denoised signal.

The prior for a detail coefficient at level j is

$$\theta_{j,k} \sim \pi_j \delta_0 + (1 - \pi_j)\{\omega_j g_W(\theta; \beta_j) + (1 - \omega_j) g_S(\theta; \beta_j)\}.$$

Here π_j is the spike probability, ω_j is the conditional Wendland weight among nonzero coefficients, and β_j is the common support scale.

See `wswavelet` for the empirical-Bayes fitting details and `level_posterior` for the coefficient-level posterior calculation.

Author(s)

Nilotpal Sanyal

References

Sanyal, N. (2026). Resolution-Adaptive Compact-Support Priors for Bayesian Wavelet Denoising: A Wendland-Semicircle Slab Mixture for Low-SNR Signal Recovery. *Axioms*, *15*(9), 678. <DOI:10.3390/axioms15090678>

See Also

`wswavelet`, `level_posterior`, `evaluate_shrinkage_curve` `estimate_noise_sd` *Robust Wavelet-Based Noise-Scale Estimation*

Description

Estimates the observational noise standard deviation from high-frequency wavelet coefficients using a robust median absolute deviation rule.

Usage

```
estimate_noise_sd(
  wavelet_object,
  detail_levels,
  supplied_sd = NULL,
  mad_levels = 1L
)
```

Arguments

`wavelet_object`	A wavelet object returned by `wd`.
`detail_levels`	A nonempty numeric or integer vector of detail levels available in `wavelet_object`. The finest levels are selected.
`supplied_sd`	Optional known positive noise standard deviation. If supplied, it is returned directly and the MAD calculation is skipped.
`mad_levels`	The positive integer number of finest detail levels to pool when estimating the scale. The default is 1.

Details

Unless `supplied_sd` is provided, the function computes

$$\widehat{\sigma} = \frac{\text{median}(|d - \text{median}(d)|)}{0.6745}$$

using the finest `mad_levels` detail levels. If the resulting value is not positive and finite, the function falls back to pooling up to the three finest available detail levels. A strictly positive machine-scale value is returned in the remaining degenerate case.

Value

A single positive numeric value giving the supplied or estimated noise standard deviation.

Author(s)

Nilotpal Sanyal

References

Sanyal, N. (2026). Resolution-Adaptive Compact-Support Priors for Bayesian Wavelet Denoising: A Wendland-Semicircle Slab Mixture for Low-SNR Signal Recovery. *Axioms*, *15*(9), 678. <DOI:10.3390/axioms15090678>

See Also

wswavelet, wd

Examples

```
set.seed(1)
y <- rnorm(128)
wd <- wavethresh::wd(y, filter.number = 6L, bc = "periodic")
lev <- 0:(wavethresh::nlevelsWT(wd) - 1L)
estimate_noise_sd(wd, lev)
```

---

evaluate_shrinkage_curve

Evaluate a Fitted Wavelet Shrinkage Curve

Description

Evaluates the fitted posterior-mean shrinkage rule and posterior component probabilities over a grid of hypothetical observed coefficients at one detail level.

Usage

```
evaluate_shrinkage_curve(
  fit,
  level_index = 1L,
  d_grid = seq(-4, 4, length.out = 401L)
)
```

Arguments

fit	A fitted object returned by wswavelet.
level_index	The position of the detail level in `fit$detail`, beginning at 1. This is the position in the fitted list, not necessarily the numerical wavelet level.
d_grid	A numeric vector of hypothetical observed coefficient values at which the shrinkage rule is evaluated.

Details

For each value in d_grid, the function recomputes the posterior for the selected level using the fitted π_j, ω_j, β_j, and likelihood. The resulting curve is the empirical-Bayes version of the coefficientwise posterior-mean rule

$$\delta_j(d) = p_{W,j}(d)\mu_{W,j}(d) + p_{S,j}(d)\mu_{S,j}(d).$$

The output can be used to plot attenuation, compare the rule with the identity map $\delta(d) = d$, or inspect how posterior evidence for the spike changes with the observed coefficient.

Value

A data frame with columns d, estimate, posterior_spike, posterior_wendland, and posterior_semicircle.

Author(s)

Nilotpal Sanyal

References

Sanyal, N. (2026). Resolution-Adaptive Compact-Support Priors for Bayesian Wavelet Denoising: A Wendland-Semicircle Slab Mixture for Low-SNR Signal Recovery. *Axioms*, *15*(9), 678. <DOI:10.3390/axioms15090678>

See Also

wswavelet, level_posterior

Examples

```
set.seed(1)
n <- 128
x <- seq(0, 1, length.out = n)
y <- sin(6 * pi * x) + rnorm(n, sd = 0.5)
fit <- wswavelet(
  y, filter.number = 6L, quadrature_n = 24L
)
curve <- evaluate_shrinkage_curve(fit, level_index = 1L)
head(curve)
```

---

level_posterior *Posterior Probabilities and Means for One Wavelet Level*

Description

Computes the complete three-component posterior for a vector of observed wavelet coefficients at one resolution level.

Usage

```
level_posterior(
  d,
  pi_j,
  omega_j,
  beta_j,
  sigma,
  likelihood,
  laplace_rate,
  quadrature_n,
  use_exact_wendland
)
```

Arguments

`d`	A numeric vector of observed wavelet coefficients. The calculation is vectorized, with one posterior calculation for each element.
`pi_j`	The prior probability of the point-mass spike at zero for the resolution level. It must lie in $[0, 1]$.
`omega_j`	The conditional prior probability of the Wendland slab among nonzero coefficients. It must lie in $[0, 1]$.
`beta_j`	The positive common support scale for the two continuous slabs.
`sigma`	The positive Gaussian noise standard deviation. It is used for the Gaussian likelihood and for Gaussian quadrature calculations.
`likelihood`	Either `"gaussian"` or `"laplace"`.
`laplace_rate`	The positive rate λ used by the Laplace working likelihood. It is ignored when `likelihood = "gaussian"`, but should still be supplied.
`quadrature_n`	The number of Gauss-Legendre nodes used for numerical integration.
`use_exact_wendland`	Logical value indicating whether the exact finite-sum Wendland formula should be used for Laplace moments.

Details

The continuous slab is

$$g_j(\theta) = \omega_j g_W(\theta; \beta_j) + (1 - \omega_j) g_S(\theta; \beta_j).$$

For component $c \in \{W, S\}$, define

$$M_{r,c,j}(d) = \int_{-\beta_j}^{\beta_j} \theta^r g_c(\theta; \beta_j) L(d \mid \theta)\, d\theta, \qquad r \in \{0, 1\}.$$

The posterior component probabilities are

$$p_{0,j}(d) = \frac{\pi_j L(d \mid 0)}{D_j(d)}, \quad p_{W,j}(d) = \frac{(1 - \pi_j)\omega_j M_{0,W,j}(d)}{D_j(d)}, \quad p_{S,j}(d) = \frac{(1 - \pi_j)(1 - \omega_j) M_{0,S,j}(d)}{D_j(d)},$$

where
$$D_j(d) = \pi_j L(d \mid 0) + (1-\pi_j)\{\omega_j M_{0,W,j}(d) + (1-\omega_j) M_{0,S,j}(d)\}.$$

The returned posterior mean is

$$\delta_j(d) = p_{W,j}(d)\mu_{W,j}(d) + p_{S,j}(d)\mu_{S,j}(d), \qquad \mu_{c,j}(d) = M_{1,c,j}(d)/M_{0,c,j}(d).$$

The spike contributes zero to this posterior mean. Computations are performed on the log scale where appropriate to reduce numerical underflow. Endpoint cases such as $\pi_j = 1$ or $\omega_j = 0$ are handled explicitly.

Value

A list with one row or one entry per value of `d`:

`log_marginal` The log marginal density $\log D_j(d)$.

`probability` A matrix with columns `spike`, `wendland`, and `semicircle`, containing posterior component probabilities.

`posterior_mean` The posterior-mean shrinkage estimate $\delta_j(d)$.

`component_mean` A matrix containing the posterior means conditional on the Wendland and semicircle components.

`component_log_m0` A matrix containing the log zeroth moments for the two continuous components.

`quadrature_fallbacks` The number of Wendland coefficients for which the exact Laplace calculation used numerical fallback.

Author(s)

Nilotpal Sanyal

References

Sanyal, N. (2026). Resolution-Adaptive Compact-Support Priors for Bayesian Wavelet Denoising: A Wendland-Semicircle Slab Mixture for Low-SNR Signal Recovery. *Axioms*, *15*(9), 678. <DOI:10.3390/axioms15090678>

See Also

`wswavelet`, `evaluate_shrinkage_curve`, `wendland_kernel`, `semicircle_kernel`

Examples

```
d <- c(-1, 0, 1)
post <- level_posterior(
  d = d,
  pi_j = 0.75,
  omega_j = 0.5,
  beta_j = 2,
  sigma = 1,
  likelihood = "gaussian",
```

```
  laplace_rate = 0.5,
  quadrature_n = 24L,
  use_exact_wendland = TRUE
)
post$posterior_mean
post$probability
```

`semicircle_kernel` *Standardized Semicircle Slab Density*

Description

Evaluates the normalized semicircle slab density on the standardized interval $[-1, 1]$.

Usage

```
semicircle_kernel(u)
```

Arguments

`u`	A numeric vector of standardized coefficient values.

Details

The density is

$$K_S(u) = \frac{2}{\pi}\sqrt{1-u^2}I(|u|<1).$$

It is symmetric, integrates to one, and is more dispersed than the Wendland component used in the package. For a support scale $\beta > 0$, the corresponding density for a coefficient θ is $g_S(\theta;\beta) = \beta^{-1}K_S(\theta/\beta)$.

Value

A numeric vector of the same length as `u`, containing the density values.

Author(s)

Nilotpal Sanyal

References

Sanyal, N. (2026). Resolution-Adaptive Compact-Support Priors for Bayesian Wavelet Denoising: A Wendland-Semicircle Slab Mixture for Low-SNR Signal Recovery. *Axioms*, *15*(9), 678. <DOI:10.3390/axioms15090678>

See Also

`wendland_kernel`, `level_posterior`

Examples

```
u <- seq(-1.2, 1.2, length.out = 7)
semicircle_kernel(u)
```

`wendland_kernel` *Standardized Wendland-Type Slab Density*

Description

Evaluates the normalized compactly supported Wendland-type slab density on the standardized interval $[-1, 1]$.

Usage

```
wendland_kernel(u)
```

Arguments

`u`	A numeric vector of standardized coefficient values.

Details

The density is

$$K_W(u) = \frac{3}{2}(1 - |u|)^4(1 + 4|u|)I(|u| < 1).$$

It is symmetric, integrates to one, is concentrated near zero, and is exactly zero outside the open unit interval. For a support scale $\beta > 0$, the corresponding density for a coefficient θ is $g_W(\theta; \beta) = \beta^{-1}K_W(\theta/\beta)$.

Value

A numeric vector of the same length as `u`, containing the density values.

Author(s)

Nilotpal Sanyal

References

Sanyal, N. (2026). Resolution-Adaptive Compact-Support Priors for Bayesian Wavelet Denoising: A Wendland-Semicircle Slab Mixture for Low-SNR Signal Recovery. *Axioms*, *15*(9), 678. <DOI:10.3390/axioms15090678>

See Also

`semicircle_kernel`, `level_posterior`

Examples

```
u <- seq(-1.2, 1.2, length.out = 7)
wendland_kernel(u)
```

---

wswavelet *Resolution-Adaptive Bayesian Wavelet Denoising*

Description

Fits the resolution-adaptive Wendland-Semicircle Bayesian wavelet denoiser, shrinks detail coefficients by posterior means, and reconstructs the signal with the inverse discrete wavelet transform.

Usage

```
wswavelet(
  y,
  filter.number = 10L,
  family = "DaubExPhase",
  bc = "periodic",
  j0 = 0L,
  likelihood = c("gaussian", "laplace"),
  supplied_sd = NULL,
  spike_offset = 2,
  spike_gamma = 2.4,
  beta_quantile = 0.99,
  beta_quantile_type = 8L,
  beta_floor = 1e-8,
  omega_model = c("adaptive", "constant", "fixed"),
  fixed_omega = NULL,
  eta_start = c(0, 0),
  eta_bounds = c(-12, 12),
  laplace_rate = NULL,
  quadrature_n = 48L,
  use_exact_wendland = TRUE,
  mad_levels = 1L,
  return_wavelet = FALSE
)
```

Arguments

`y`	A finite numeric signal vector. Its length must be a dyadic integer, such as 128, 256, or 1024.
`filter.number`	The wavelet filter order passed to `wd`. The default is 10.
`family`	The wavelet family passed to `wd`. Common choices include `"DaubExPhase"` and `"DaubLeAsymm"`.
`bc`	The boundary convention passed to `wd`. The default is `"periodic"`.

`j0`	The coarsest detail level to shrink. Detail levels below `j0` are not shrunk. The default is 0.
`likelihood`	The coefficientwise working likelihood. Use `"gaussian"` for the primary version or `"laplace"` for the robustness-oriented alternative.
`supplied_sd`	Optional known noise standard deviation. If `NULL`, the function estimates it from high-frequency wavelet coefficients using a robust MAD rule.
`spike_offset`	The offset ℓ in the level-dependent spike probability. The implementation permits values at least 1; the default is 2.
`spike_gamma`	The positive resolution-sparsity exponent γ. Larger values make the spike probability increase more rapidly with resolution.
`beta_quantile`	The probability κ used in the empirical upper-quantile rule for the support scale β_j. It must lie strictly between 0 and 1; the default is 0.99.
`beta_quantile_type`	The integer quantile convention passed to `quantile`. The default is type 8.
`beta_floor`	A positive numerical floor used to prevent a zero support scale when the noise estimate and coefficients are numerically zero.
`omega_model`	How the Wendland weight is modeled across resolution levels. `"adaptive"` estimates an intercept and a resolution trend, `"constant"` estimates one common weight, and `"fixed"` uses `fixed_omega`.
`fixed_omega`	A fixed Wendland weight in $[0, 1]$. Supplying this argument forces `omega_model = "fixed"`; 0 gives the semicircle-only endpoint and 1 gives the Wendland-only endpoint.
`eta_start`	Starting values for the empirical-Bayes logistic trend parameters (η_0, η_1). For a constant mixture, only the first value is used.
`eta_bounds`	Common finite lower and upper bounds for the optimized logistic parameters. The default is `c(-12, 12)`.
`laplace_rate`	Optional positive rate parameter for the Laplace working likelihood. If `NULL`, the function sets $\widehat{\lambda} = 1/(2\widehat{\sigma}^2)$ once before optimization and holds it fixed. This argument has no effect for the Gaussian likelihood.
`quadrature_n`	Number of Gauss-Legendre nodes used for numerical integration. The default is 48. Larger values may improve accuracy but increase computation time.
`use_exact_wendland`	Logical value. For the Laplace likelihood, `TRUE` uses the finite-sum Wendland moment formula with a numerical fallback. If `FALSE`, numerical quadrature is used. For the Gaussian likelihood, numerical quadrature is used.
`mad_levels`	Number of finest detail levels pooled for the robust noise-scale estimate. The default is 1.
`return_wavelet`	Logical value. If `TRUE`, also returns the original and thresholded `wavethresh` wavelet objects.

Details

The prior for a detail coefficient at resolution level j is

$$\theta_{j,k} \sim \pi_j \delta_0 + (1 - \pi_j)\{\omega_j g_W(\theta; \beta_j) + (1 - \omega_j) g_S(\theta; \beta_j)\}.$$

The spike probability is

$$\pi_j = 1 - (j - j_0 + \ell)^{-\gamma}.$$

For the adaptive model, the Wendland weight is

$$\omega_j(\eta) = \operatorname{logit}^{-1}(\eta_0 + \eta_1 t_j), \qquad t_j = (j - j_0)/(J - 1 - j_0).$$

The support scale is selected from the observed detail coefficients using

$$\widehat{\beta}_j = \max\{\widehat{\sigma}, Q_\kappa(|d_{j,k}|)\},$$

followed by the positive numerical floor

$$\beta_{\min} = \epsilon_\beta s_\beta, \qquad s_\beta = \max\{1, \widehat{\sigma}, \max_{j,k} |d_{j,k}|\}, \qquad \widehat{\beta}_j \leftarrow \max\{\widehat{\beta}_j, \beta_{\min}\},$$

where ϵ_β is `beta_floor`.

When `omega_model = "constant"`, the fitted Wendland weight is $\operatorname{logit}^{-1}(\eta_0)$ at every level. When `omega_model = "fixed"`, no weight optimization is performed.

For Gaussian errors, the coefficientwise likelihood is

$$L_N(d \mid \theta, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp\{-(d - \theta)^2/(2\sigma^2)\}.$$

For the Laplace working likelihood, the implementation uses

$$L_L(d \mid \theta, \lambda) = (a/2) \exp\{-a|d - \theta|\}, \qquad a = \sqrt{2\lambda}.$$

The empirical-Bayes parameters are fitted by bounded L-BFGS-B optimization of the joint log marginal likelihood. The posterior mean replaces each detail coefficient, while the scaling coefficients are retained without shrinkage.

Value

A list with the following principal components:

`estimate` The reconstructed denoised signal.

`sigma_hat` The estimated or supplied noise standard deviation.

`laplace_rate_hat` The rate used for the Laplace working likelihood.

`likelihood` The selected likelihood.

`omega_model` The selected mixture-weight model.

`eta_hat` The fitted logistic-trend parameters, padded to length two for the constant and fixed cases.

`optimization` The optimization result or fixed-weight record.

`level_summary` A data frame containing level-specific support scales, prior probabilities, fitted weights, average posterior component probabilities, and numerical diagnostics.

`detail` A list containing the observed and fitted coefficients and posterior quantities for each detail level.

`original` The input signal.

`diagnostics` Quadrature, fallback, non-finite-integral, quantile-type, and MAD-level information.

`settings` The principal fitting settings.

If `return_wavelet = TRUE`, the list also contains `wavelet_object` and `thresholded_wavelet`.

Author(s)

Nilotpal Sanyal

References

Sanyal, N. (2026). Resolution-Adaptive Compact-Support Priors for Bayesian Wavelet Denoising: A Wendland-Semicircle Slab Mixture for Low-SNR Signal Recovery. *Axioms*, *15*(9), 678. <DOI:10.3390/axioms15090678>

See Also

level_posterior, evaluate_shrinkage_curve, estimate_noise_sd, wd

Examples

```
set.seed(1)
n <- 128
x <- seq(0, 1, length.out = n)
y <- sin(6 * pi * x) + rnorm(n, sd = 0.5)

fit <- wswavelet(
  y,
  likelihood = "gaussian",
  filter.number = 6L,
  quadrature_n = 24L
)
fit$estimate
fit$level_summary
```

---

ws_numerical_checks *Numerical Checks for the WS Posterior-Mean Rule*

Description

Runs a standard collection of numerical checks for endpoint handling, symmetry, support bounds, finiteness, and Gaussian monotonicity of the posterior-mean shrinkage rule.

Usage

```
ws_numerical_checks(quadrature_n = 128L)
```

Arguments

quadrature_n	The number of Gauss-Legendre nodes used in the numerical integrations. Larger values can improve accuracy but increase computation time.

Details

The routine evaluates Gaussian and Laplace posterior means over a fixed coefficient grid for all combinations of spike probability $\pi \in \{0, 0.5, 1\}$ and Wendland weight $\omega \in \{0, 0.5, 1\}$. It checks the numerical versions of

$$\delta(-d) = -\delta(d)$$

and

$$|\delta(d)| \leq \beta$$

. For the Gaussian likelihood, `min_first_difference` checks the expected nondecreasing behavior on the grid. The endpoint columns verify the exact no-spike and all-spike rules.

Value

A data frame with one row for each likelihood, spike probability, and slab weight combination. Important columns are `max_oddness_error`, `min_first_difference`, `max_support_excess`, `all_finite`, `pi0_exact_slab`, and `pi1_exact_zero`.

Author(s)

Nilotpal Sanyal

References

Sanyal, N. (2026). Resolution-Adaptive Compact-Support Priors for Bayesian Wavelet Denoising: A Wendland-Semicircle Slab Mixture for Low-SNR Signal Recovery. *Axioms*, *15*(9), 678. <DOI:10.3390/axioms15090678>

See Also

`level_posterior`, `wswavelet`

Examples

```
checks <- ws_numerical_checks(quadrature_n = 48L)
checks
```

Index