Package ‘powerprior’

November 11, 2025

Version 1.0.0

Title Conjugate Power Priors for Bayesian Analysis of Normal Data

Description Implements conjugate power priors for efficient Bayesian analysis of normal data. Power priors allow principled incorporation of historical information while controlling the degree of borrowing through a discounting parameter (Ibrahim and Chen (2000) <doi:10.1214/ss/1009212519>). This package provides closed-form conjugate representations for both univariate and multivariate normal data using Normal-Inverse-Chi-squared and Normal-Inverse-Wishart distributions, eliminating the need for MCMC sampling. The conjugate framework builds upon standard Bayesian methods described in Gelman et al. (2013, ISBN:978-1439840955).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats, MASS, LaplacesDemon, ggplot2, shiny, shinydashboard, shinyjs, DT, dplyr, tidyr, rlang

Suggests testthat (>= 3.0.0), rmarkdown

NeedsCompilation no

Author Yusuke Yamaguchi [aut, cre], Yifei Huang [aut]

Maintainer Yusuke Yamaguchi <yamagubed@gmail.com>

Repository CRAN

Date/Publication 2025-11-11 21:40:13 UTC

Contents

`bayes_factor` *Calculate Bayes Factor*

Description

Computes the Bayes factor comparing two models with different discounting parameters.

Usage

```
bayes_factor(
  historical_data,
  current_data,
  a0_1,
  a0_2 = 0,
  multivariate = FALSE,
  ...
)
```

Arguments

`historical_data`	Historical data
`current_data`	Current data
`a0_1`	First discounting parameter
`a0_2`	Second discounting parameter (default: 0)
`multivariate`	Logical indicating multivariate data (default: FALSE)
`...`	Additional arguments passed to powerprior functions

Details

The Bayes factor compares the marginal likelihoods under two different discounting parameters. BF > 1 favors a0_1, BF < 1 favors a0_2.

Note: This is a simple approximation using the observed data likelihood evaluated at posterior means.

Value

Bayes factor (BF_12 = P(data|a0_1) / P(data|a0_2))

Examples

```
set.seed(123)
historical <- rnorm(50, mean = 10, sd = 2)
current <- rnorm(30, mean = 10.5, sd = 2)

# Compare moderate borrowing (0.5) vs no borrowing (0)
bf <- bayes_factor(historical, current, a0_1 = 0.5, a0_2 = 0)
cat("Bayes Factor:", bf, "\n")
```

---

check_compatibility *Check Data Compatibility*

Description

Performs a simple compatibility check between historical and current data to help guide the choice of discounting parameter.

Usage

```
check_compatibility(historical_data, current_data, alpha = 0.05)
```

Arguments

`historical_data`	Historical data
`current_data`	Current data
`alpha`	Significance level for compatibility test (default: 0.05)

Value

A list with compatibility test results

Examples

```
set.seed(123)
historical <- rnorm(50, mean = 10, sd = 2)
current <- rnorm(30, mean = 10.5, sd = 2)

compatibility <- check_compatibility(historical, current)
print(compatibility)
```

---

compare_discounting *Compare Multiple Discounting Parameters*

Description

Computes posteriors for multiple values of a0 to facilitate sensitivity analysis.

Usage

```
compare_discounting(
  historical_data,
  current_data,
  a0_values = seq(0, 1, by = 0.1),
  multivariate = FALSE,
  n_samples = 1000,
  ...
)
```

Arguments

`historical_data`	Historical data (vector for univariate, matrix for multivariate)
`current_data`	Current data (vector for univariate, matrix for multivariate)
`a0_values`	Vector of discounting parameters to compare
`multivariate`	Logical indicating if data is multivariate (default: FALSE)
`n_samples`	Number of posterior samples per a0 value (default: 1000)
`...`	Additional arguments passed to powerprior functions

Value

A data frame with summary statistics for each a0 value

Examples

```
set.seed(123)
historical <- rnorm(50, mean = 10, sd = 2)
current <- rnorm(30, mean = 10.5, sd = 2)

comparison <- compare_discounting(
  historical,
  current,
  a0_values = c(0, 0.25, 0.5, 0.75, 1.0)
)
print(comparison)
```

`effective_sample_size` *Calculate Effective Sample Size*

Description

Computes the effective sample size from historical data given a discounting parameter.

Usage

```
effective_sample_size(m, a0)
```

Arguments

`m`	Sample size of historical data
`a0`	Discounting parameter

Value

Effective sample size (numeric)

Examples

```
# With 100 historical observations and 50% discounting
effective_sample_size(100, a0 = 0.5)  # Returns 50
```

plot_sensitivity *Plot Sensitivity Analysis*

Description

Creates a plot showing how posterior estimates vary with the discounting parameter.

Usage

```
plot_sensitivity(comparison_results, parameter = "mu", dimension = 1)
```

Arguments

comparison_results	Output from compare_discounting()
parameter	Name of parameter to plot (default: "mu")
dimension	For multivariate case, which dimension to plot (default: 1)

Value

A ggplot2 object (if ggplot2 is available) or base R plot

Examples

```
set.seed(123)
historical <- rnorm(50, mean = 10, sd = 2)
current <- rnorm(30, mean = 10.5, sd = 2)

comparison <- compare_discounting(historical, current)
plot_sensitivity(comparison)
```

posterior_multivariate *Posterior Obtained by Updating Power Prior with Current Data (Multivariate)*

Description

Updates a multivariate power prior with current trial data to obtain the posterior distribution.

Usage

```
posterior_multivariate(powerprior, current_data)
```

Arguments

`powerprior` Object of class "powerprior_multivariate" from powerprior_multivariate()

`current_data` Matrix or data frame of current observations. Must have the same number of columns as the historical data used to create the power prior. Rows represent observations, columns represent variables.

Details

Posterior Updating in the Multivariate Setting:

Given a power prior P(μ, Σ | X, a_0) and new current data Y, the posterior distribution combines both through Bayes' theorem. The conjugate NIW structure ensures the posterior remains a NIW distribution with closed-form parameters.

The updating mechanism mirrors the univariate case but extends to handle the covariance matrix structure. The scale matrix Λ^* incorporates:

1. Discounted historical sum of squares and crossproducts (a_0 * S_x)
2. Prior scale information (Λ_0, if using informative prior)
3. Current data sum of squares and crossproducts (S_y)
4. Disagreement terms that increase uncertainty when historical and current means differ

The posterior mean vector is a precision-weighted average of the historical mean, prior mean (if provided), and current mean, allowing for flexible incorporation of multiple information sources with different precisions.

Value

A list of class "posterior_multivariate" containing:

`mu_star` Posterior mean vector

`kappa_star` Posterior precision parameter

`nu_star` Posterior degrees of freedom

`Lambda_star` Posterior scale matrix

`n` Sample size of current data

`p` Dimension of data

`ybar` Sample mean vector of current data

`Sy` Sum of squares and crossproducts matrix for current data

`powerprior` Original power prior object

Examples

```
library(MASS)
Sigma_true <- matrix(c(4, 1, 1, 2), 2, 2)
historical <- mvrnorm(50, mu = c(10, 5), Sigma = Sigma_true)
current <- mvrnorm(30, mu = c(10.5, 5.2), Sigma = Sigma_true)

# With vague prior
pp <- powerprior_multivariate(historical, a0 = 0.5)
```

```
posterior <- posterior_multivariate(pp, current)
print(posterior)

# With informative prior
pp_inform <- powerprior_multivariate(
  historical, a0 = 0.5,
  mu0 = c(10, 5), kappa0 = 1, nu0 = 5, Lambda0 = diag(2)
)
posterior_inform <- posterior_multivariate(pp_inform, current)
print(posterior_inform)
```

---

posterior_summary *Compute Posterior Summaries*

Description

Provides comprehensive posterior summary statistics.

Usage

```
posterior_summary(posterior, n_samples = 10000, prob = 0.95)
```

Arguments

posterior	Posterior object from posterior_univariate() or posterior_multivariate()
n_samples	Number of samples to draw (default: 10000)
prob	Probability for credible intervals (default: 0.95)

Value

A list with posterior summaries

Examples

```
set.seed(123)
historical <- rnorm(50, mean = 10, sd = 2)
current <- rnorm(30, mean = 10.5, sd = 2)

pp <- powerprior_univariate(historical, a0 = 0.5)
posterior <- posterior_univariate(pp, current)
summary <- posterior_summary(posterior)
print(summary)
```

`posterior_univariate` *Posterior Obtained by Updating Power Prior with Current Data (Univariate)*

Description

Updates a power prior with current trial data to obtain the posterior distribution.

Usage

```
posterior_univariate(powerprior, current_data)
```

Arguments

`powerprior`	Object of class "powerprior_univariate" from powerprior_univariate()
`current_data`	Numeric vector of current trial observations. Must contain at least 2 observations. Missing values (NAs) are automatically removed.

Details

Posterior Updating:

Given a power prior distribution P(μ, σ^2 | x, a_0) and new current data y, the posterior distribution is computed by combining the power prior with the likelihood of the current data using Bayes' theorem.

The conjugate structure ensures the posterior remains a NIX distribution. For both informative and vague initial priors, the updating follows standard conjugate rules, leveraging the fact that both the power prior and likelihood are in the NIX family.

This eliminates the computational burden of MCMC and allows direct posterior inference and sampling (see `sample_posterior_univariate`).

Value

A list of class "posterior_univariate" containing:

`mu_star`	Posterior mean parameter
`kappa_star`	Posterior precision parameter
`nu_star`	Posterior degrees of freedom
`sigma2_star`	Posterior scale parameter
`n`	Sample size of current data
`ybar`	Sample mean of current data
`Sy`	Sum of squared deviations of current data
`powerprior`	Original power prior object

Examples

```
# Generate data
historical <- rnorm(50, mean = 10, sd = 2)
current <- rnorm(30, mean = 10.5, sd = 2)

# Compute power prior and posterior
pp <- powerprior_univariate(historical, a0 = 0.5)
posterior <- posterior_univariate(pp, current)
print(posterior)

# With informative prior
pp_inform <- powerprior_univariate(
  historical, a0 = 0.5,
  mu0 = 10, kappa0 = 1, nu0 = 3, sigma2_0 = 4
)
posterior_inform <- posterior_univariate(pp_inform, current)
print(posterior_inform)
```

`powerprior_launch_shinyapp`
Launch the Shiny App

Description

This function launches the Shiny application.

Usage

```
powerprior_launch_shinyapp()
```

Value

No return value, called for side effects (launches Shiny app)

`powerprior_multivariate`
Compute Power Prior for Multivariate Normal Data

Description

Computes the power prior for multivariate normal data using a conjugate Normal-Inverse-Wishart (NIW) representation.

Usage

```
powerprior_multivariate(
  historical_data,
  a0,
  mu0 = NULL,
  kappa0 = NULL,
  nu0 = NULL,
  Lambda0 = NULL
)
```

Arguments

`historical_data`
: Matrix or data frame where rows are observations and columns are variables. Must have at least 2 rows and 2 columns. Missing values are not automatically handled; rows with any NA values should be removed before calling this function.

`a0`
: Discounting parameter in `[0, 1]`. Controls the degree of borrowing from historical data in the multivariate setting. Same interpretation as univariate case:
  - a0 = 0: No borrowing from historical data
  - a0 = 0.5: Moderate borrowing with half weight
  - a0 = 1: Full borrowing of historical information

`mu0`
: Prior mean vector of length p (number of variables). Only used when specifying an informative initial prior. If NULL, a vague (non-informative) prior is used. Represents the prior belief about the center of the multivariate distribution across all variables.

`kappa0`
: Prior precision parameter (scalar). Controls the concentration of the prior around mu0. Only used for informative priors. If NULL, vague prior is applied. Interpreted as the "prior effective sample size" for the mean estimate. Higher values indicate greater prior confidence.

`nu0`
: Prior degrees of freedom for the inverse Wishart distribution. Only used for informative priors. If NULL, vague prior is applied. Must satisfy nu0 >= p. Higher values indicate greater prior confidence in the covariance structure. Typical values range from p to 2p for informative priors.

`Lambda0`
: Prior scale matrix (p x p, symmetric positive definite). Only used for informative priors. If NULL, vague prior is applied. Represents the prior belief about the covariance structure. Larger values correspond to greater prior uncertainty about variable spread and relationships. If Lambda0 is provided, must be symmetric and positive definite.

Details

Background on Multivariate Power Priors:

The power prior framework extends naturally to multivariate normal data when the mean vector μ and covariance matrix Σ are jointly unknown. This is essential for modern applications involving multiple correlated endpoints, such as clinical trials measuring multiple health outcomes simultaneously.

The power prior for multivariate data is defined analogously to the univariate case:

$$P(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X}, a_0) = L(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X})^{a_0} P(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where:

- $\mathbf{X}$ is the m x p historical data matrix
- $\boldsymbol{\mu}$ is the p-dimensional mean vector
- $\boldsymbol{\Sigma}$ is the p x p covariance matrix
- $a_0 \in [0, 1]$ is the discounting parameter

Conjugacy via Normal-Inverse-Wishart Distribution:

The key advantage of this implementation is that power priors applied to multivariate normal data with Normal-Inverse-Wishart (NIW) conjugate initial priors remain in closed form as NIW distributions. This preserves:

- Exact posterior computation without approximation
- Closed-form parameter updates and marginalization
- Efficient sampling from standard distributions
- Computational tractability for high-dimensional problems (within reason)
- Natural joint modeling of correlations via the covariance structure

For practitioners, this means you can incorporate historical information on multiple correlated endpoints while maintaining full Bayesian rigor and computational efficiency.

Informative vs. Vague Priors:

Informative Initial Prior (all of mu0, kappa0, nu0, Lambda0 provided):

Uses a Normal-Inverse-Wishart (NIW) conjugate prior with hierarchical structure:

$$\boldsymbol{\mu}|\boldsymbol{\Sigma} \sim N_p(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}/\kappa_0)$$
$$\boldsymbol{\Sigma} \sim \text{Inv-Wishart}(\nu_0, \boldsymbol{\Lambda}_0)$$

The power prior parameters are updated:

$$\boldsymbol{\mu}_n = \frac{a_0 m \overline{\mathbf{x}} + \kappa_0 \boldsymbol{\mu}_0}{a_0 m + \kappa_0}$$

$$\kappa_n = a_0 m + \kappa_0$$

$$\nu_n = a_0 m + \nu_0$$

$$\boldsymbol{\Lambda}_n = a_0 \mathbf{S}_x + \boldsymbol{\Lambda}_0 + \frac{\kappa_0 a_0 m}{a_0 m + \kappa_0} (\boldsymbol{\mu}_0 - \overline{\mathbf{x}})(\boldsymbol{\mu}_0 - \overline{\mathbf{x}})^T$$

where m is sample size, $\overline{\mathbf{x}}$ is the sample mean vector, and $\mathbf{S}_x = \sum_{i=1}^m (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T$ is the sum of squares and crossproducts matrix.

Vague (Non-informative) Initial Prior (all of mu0, kappa0, nu0, Lambda0 are NULL):

Uses a non-informative prior $P(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{-(p+1)/2}$ that places minimal constraints on parameters. The power prior parameters simplify to:

$$\boldsymbol{\mu}_n = \bar{\mathbf{x}}$$

$$\kappa_n = a_0 m$$

$$\nu_n = a_0 m - 1$$

$$\boldsymbol{\Lambda}_n = a_0 \mathbf{S}_x$$

The vague prior is recommended when there is minimal prior information, or when you want the analysis driven primarily by the historical data.

Parameter Interpretation in Multivariate Setting:

Effective Sample Size (κ_n): Quantifies how much "effective historical data" has been incorporated. The formula $\kappa_n = a_0 m + \kappa_0$ shows the discounted historical sample size combined with prior precision. This controls the concentration of the posterior distribution for the mean vector.

Mean Vector (μ_n): The updated mean is a precision-weighted average: $\boldsymbol{\mu}_n = \frac{a_0 m \bar{\mathbf{x}} + \kappa_0 \boldsymbol{\mu}_0}{a_0 m + \kappa_0}$ This naturally balances the historical sample mean and prior mean, with weights proportional to their respective precisions.

Degrees of Freedom (ν_n): Controls tail behavior and the concentration of the Wishart distribution. Higher values indicate greater confidence in the covariance estimate. The minimum value needed is p (number of variables) for the Inverse-Wishart to be well-defined; $\nu_n \geq p$ is always maintained.

Scale Matrix (Λ_n): The p x p scale matrix that captures both the dispersion of individual variables and their correlations. The term $\frac{\kappa_0 a_0 m}{a_0 m + \kappa_0}(\boldsymbol{\mu}_0 - \bar{\mathbf{x}})(\boldsymbol{\mu}_0 - \bar{\mathbf{x}})^T$ adds uncertainty when the historical mean conflicts with the prior mean, naturally reflecting disagreement between data sources.

Practical Considerations:

Dimension: This implementation works efficiently for moderate-dimensional problems (typically p $\leq$ 10). For higher dimensions, consider data reduction techniques or structural assumptions on the covariance matrix.

Prior Specification: When specifying Lambda0, ensure it is symmetric positive definite. A simple approach is to use a multiple of the identity matrix (e.g., Lambda0 = diag(p)) for a weakly informative prior.

Discounting: The same a0 parameter is used for all variables and their correlations. If you suspect differential reliability of historical information across variables, consider multiple analyses with different a0 values and sensitivity analyses.

Value

A list of class "powerprior_multivariate" containing:

`mu_n`	Updated mean vector (p-dimensional)
`kappa_n`	Updated precision parameter (effective sample size)
`nu_n`	Updated degrees of freedom
`Lambda_n`	Updated scale matrix (p x p)

a0	Discounting parameter used
m	Sample size of historical data
p	Dimension (number of variables) of the data
xbar	Sample mean vector of historical data
Sx	Sum of squares and crossproducts matrix
vague_prior	Logical indicating if vague prior was used
mu0	Prior mean vector (if informative prior used)
kappa0	Prior precision parameter (if informative prior used)
nu0	Prior degrees of freedom (if informative prior used)
Lambda0	Prior scale matrix (if informative prior used)

References

Huang, Y., Yamaguchi, Y., Homma, G., Maruo, K., & Takeda, K. (2024). "Conjugate Representation of Power Priors for Efficient Bayesian Analysis of Normal Data." (unpublished).

Ibrahim, J. G., & Chen, M. H. (2000). "Power prior distributions for regression models." Statistical Science, 15(1), 46-60.

Gelman, A., Carlin, J. B., Stern, H. S., et al. (2013). Bayesian Data Analysis (3rd ed.). CRC Press.

Examples

```
# Generate multivariate historical data with correlation
library(MASS)
Sigma_true <- matrix(c(4, 1, 1, 2), 2, 2)
historical <- mvrnorm(50, mu = c(10, 5), Sigma = Sigma_true)

# Compute power prior with vague prior
pp <- powerprior_multivariate(historical, a0 = 0.5)
print(pp)

# Compute power prior with informative prior
pp_inform <- powerprior_multivariate(
  historical,
  a0 = 0.5,
  mu0 = c(10, 5),
  kappa0 = 1,
  nu0 = 5,
  Lambda0 = diag(2)
)
print(pp_inform)
```

`powerprior_univariate` *Compute Power Prior for Univariate Normal Data*

Description

Computes the power prior for univariate normal data using a conjugate Normal-Inverse-Chi-squared (NIX) representation.

Usage

```
powerprior_univariate(
  historical_data,
  a0,
  mu0 = NULL,
  kappa0 = NULL,
  nu0 = NULL,
  sigma2_0 = NULL
)
```

Arguments

`historical_data`
: Numeric vector of historical observations. Must contain at least 2 observations. Missing values (NAs) are automatically removed. The function assumes data are independent and identically distributed from a normal distribution with unknown mean and variance.

`a0`
: Discounting parameter in `[0, 1]`. Controls the degree of borrowing from historical data. Specific values have intuitive interpretations:

  - a0 = 0: No borrowing from historical data; power prior equals the initial prior
  - a0 = 0.5: Moderate borrowing; historical data contribute with half weight
  - a0 = 1: Full borrowing; historical data weighted equally with current data

  The parameter acts as a multiplicative discount on the historical likelihood.

`mu0`
: Prior mean for the normal distribution of the mean parameter. Only used when specifying an informative initial prior. If NULL, a vague (non-informative) prior is used instead. Represents the prior belief about the center of the data distribution.

`kappa0`
: Prior precision parameter (inverse of scaled variance) for the mean. Only used for informative priors. If NULL, vague prior is applied. Higher values indicate greater prior confidence in mu0. Interpreted as the "prior sample size" contributing to the mean estimate. For example, kappa0 = 1 is roughly equivalent to one prior observation.

`nu0`
: Prior degrees of freedom for the inverse chi-squared distribution of variance. Only used for informative priors. If NULL, vague prior is applied. Higher values indicate greater prior confidence in sigma2_0. Typically set to small positive values (e.g., 1-5) for weakly informative priors.

`sigma2_0` Prior scale parameter for the inverse chi-squared distribution. Only used for informative priors. If NULL, vague prior is applied. Represents the prior belief about the scale (spread) of the data. Should be on the variance scale (not standard deviation).

Details

Background on Power Priors:

Power priors provide a principled framework for incorporating historical information in Bayesian analysis while controlling the degree of borrowing through a discounting parameter. The power prior is defined as:

$$P(\theta|x, a_0) = L(\theta|x)^{a_0} P(\theta)$$

where

- $L(\theta|x)$ is the likelihood function evaluated on historical data x
- $a_0 \in [0, 1]$ is the discounting parameter
- $P(\theta)$ is the initial prior distribution

This approach is especially valuable in clinical trial design, where historical trial data can improve statistical efficiency while maintaining appropriate skepticism about whether historical and current populations are similar.

Conjugacy and Computational Efficiency:

Typically, power priors result in non-closed-form posterior distributions requiring computationally expensive Markov Chain Monte Carlo (MCMC) sampling, especially problematic for simulation studies requiring thousands of posterior samples.

This implementation exploits a key mathematical result: when applying power priors to normal data with conjugate initial priors (Normal-Inverse-Chi-squared), the resulting power prior and posterior distributions remain in closed form as NIX distributions. This allows:

- Direct computation without MCMC approximation
- Closed-form parameter updates
- Exact posterior sampling from standard distributions
- Efficient sensitivity analyses across different a0 values

Informative vs. Vague Priors:

Informative Initial Prior (all of mu0, kappa0, nu0, sigma2_0 provided):

Uses a Normal-Inverse-Chi-squared (NIX) conjugate prior with structure:

$$\mu|\sigma^2 \sim N(\mu_0, \sigma^2/\kappa_0)$$
$$\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$$

The power prior parameters are updated:

$$\mu_n = \frac{a_0 m\bar{x} + \kappa_0\mu_0}{a_0 m + \kappa_0}$$

$$\kappa_n = a_0 m + \kappa_0$$

$$\nu_n = a_0 m + \nu_0$$

$$\sigma_n^2 = \frac{1}{\nu_n}\left[a_0 S_x + \nu_0 \sigma_0^2 + \frac{a_0 m \kappa_0 (\bar{x} - \mu_0)^2}{a_0 m + \kappa_0}\right]$$

where m is the sample size, $\bar{x}$ is the sample mean, and $S_x = \sum_{i=1}^{m}(x_i - \bar{x})^2$ is the sum of squared deviations.

Vague (Non-informative) Initial Prior (all of mu0, kappa0, nu0, sigma2_0 are NULL): Uses a non-informative prior $P(\mu, \sigma^2) \propto \sigma^{-2}$ that is locally uniform in log-space. This places minimal prior constraints on the parameters. The power prior parameters simplify to:

$$\mu_n = \bar{x}$$

$$\kappa_n = a_0 m$$

$$\nu_n = a_0 m - 1$$

$$\sigma_n^2 = \frac{a_0 S_x}{\nu_n}$$

The vague prior approach is recommended when there is no strong prior information, or when you want the analysis to be primarily driven by the (discounted) historical data.

Parameter Interpretation:

Effective Sample Size (kappa_n): The updated precision parameter can be interpreted as an effective sample size. Higher values indicate more concentrated posterior distributions for the mean. The formula $\kappa_n = a_0 m + \kappa_0$ shows that the historical sample size m is discounted by a_0 before combining with the prior's contribution κ_0.

Posterior Mean (mu_n): A weighted average of the historical sample mean $\bar{x}$, prior mean μ_0, and their relative precision: $\mu_n = \frac{a_0 m \bar{x} + \kappa_0 \mu_0}{a_0 m + \kappa_0}$ This naturally interpolates between the data and prior, with weights determined by their precision.

Degrees of Freedom (nu_n): Controls the tail behavior of posterior distributions derived from the NIX. Higher values produce lighter tails, indicating greater confidence.

Scale Parameter (sigma2_n): Estimates the variability in the data. The term involving $(\bar{x} - \mu_0)^2$ captures disagreement between the historical mean and prior mean, which increases uncertainty in variance estimation when they conflict.

Value

A list of class "powerprior_univariate" containing:

`mu_n`	Updated posterior mean parameter from power prior
`kappa_n`	Updated posterior precision parameter (effective sample size)
`nu_n`	Updated posterior degrees of freedom
`sigma2_n`	Updated posterior scale parameter (variance scale)

a0	Discounting parameter used
m	Sample size of historical data
xbar	Sample mean of historical data
Sx	Sum of squared deviations of historical data
vague_prior	Logical indicating if vague prior was used
mu0	Prior mean (if informative prior used)
kappa0	Prior precision (if informative prior used)
nu0	Prior degrees of freedom (if informative prior used)
sigma2_0	Prior scale parameter (if informative prior used)

References

Huang, Y., Yamaguchi, Y., Homma, G., Maruo, K., & Takeda, K. (2024). "Conjugate Representation of Power Priors for Efficient Bayesian Analysis of Normal Data." Statistical Science (unpublished).

Ibrahim, J. G., & Chen, M. H. (2000). "Power prior distributions for regression models." Statistical Science, 15(1), 46-60.

Gelman, A., Carlin, J. B., Stern, H. S., et al. (2013). Bayesian Data Analysis (3rd ed.). CRC Press.

Examples

```
# Generate historical data
historical <- rnorm(50, mean = 10, sd = 2)

# Compute power prior with informative initial prior
pp_inform <- powerprior_univariate(
  historical_data = historical,
  a0 = 0.5,
  mu0 = 10,
  kappa0 = 1,
  nu0 = 3,
  sigma2_0 = 4
)
print(pp_inform)

# Compute power prior with vague prior (no prior specification)
pp_vague <- powerprior_univariate(
  historical_data = historical,
  a0 = 0.5
)
print(pp_vague)

# Compare different discounting levels
pp_weak <- powerprior_univariate(historical_data = historical, a0 = 0.1)
pp_strong <- powerprior_univariate(historical_data = historical, a0 = 0.9)
cat("Strong discounting (a0=0.1) - kappa_n:", pp_weak$kappa_n, "\n")
cat("Weak discounting (a0=0.9) - kappa_n:", pp_strong$kappa_n, "\n")
```

print.compatibility_check
Print method for compatibility_check

Description

Print method for compatibility_check

Usage

```
## S3 method for class 'compatibility_check'
print(x, digits = 4, ...)
```

Arguments

`x`	Object of class "compatibility_check"
`digits`	Number of digits to round to
`...`	Additional arguments

Value

Invisibly returns the input object (for method chaining)

print.posterior_multivariate
Print method for posterior_multivariate

Description

Print method for posterior_multivariate

Usage

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

Arguments

`x`	Object of class "posterior_multivariate"
`...`	Additional arguments

Value

Invisibly returns the input object (for method chaining)

print.posterior_univariate

Print method for posterior_univariate

Description

Print method for posterior_univariate

Usage

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

Arguments

`x`	Object of class "posterior_univariate"
`...`	Additional arguments

Value

Invisibly returns the input object (for method chaining)

print.powerprior_comparison

Print method for powerprior_comparison

Description

Print method for powerprior_comparison

Usage

```
## S3 method for class 'powerprior_comparison'
print(x, digits = 4, ...)
```

Arguments

`x`	Object of class "powerprior_comparison"
`digits`	Number of digits to round to
`...`	Additional arguments

Value

Invisibly returns the input object (for method chaining)

`print.powerprior_multivariate`
Print method for powerprior_multivariate

Description

Print method for powerprior_multivariate

Usage

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

Arguments

`x`	Object of class "powerprior_multivariate"
`...`	Additional arguments

Value

Invisibly returns the input object (for method chaining)

`print.powerprior_summary`
Print method for powerprior_summary

Description

Print method for powerprior_summary

Usage

```
## S3 method for class 'powerprior_summary'
print(x, digits = 4, ...)
```

Arguments

`x`	Object of class "powerprior_summary"
`digits`	Number of digits to round to
`...`	Additional arguments

Value

Invisibly returns the input object (for method chaining)

`print.powerprior_univariate`
Print method for powerprior_univariate

Description

Print method for powerprior_univariate

Usage

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

Arguments

`x`	Object of class "powerprior_univariate"
`...`	Additional arguments

Value

Invisibly returns the input object (for method chaining)

`sample_posterior_multivariate`
Sample from Posterior Distribution (Multivariate)

Description

Generates samples from the multivariate posterior distribution using exact closed-form expressions from the Normal-Inverse-Wishart conjugate family.

Usage

```
sample_posterior_multivariate(posterior, n_samples = 1000, marginal = FALSE)
```

Arguments

`posterior`	Object of class "posterior_multivariate" from posterior_multivariate()
`n_samples`	Number of samples to generate (default: 1000). For large n_samples or high dimensions, computation time increases.
`marginal`	Logical. If TRUE, samples only μ from the multivariate t-distribution. If FALSE (default), samples the joint (μ, Σ) from the NIW distribution, which is more computationally intensive but provides uncertainty in the covariance structure.

Details

Sampling Algorithms:

Joint Sampling (marginal=FALSE):

Implements the standard hierarchical sampling algorithm for the NIW distribution:

1. Draw Σ ~ Inverse-Wishart(ν^*, Λ^*)
2. Draw μ | Σ ~ N_p(μ^*, Σ / κ^*)

This produces samples from the joint distribution P(μ, Σ | Y, X, a_0) and captures both uncertainty in the mean and uncertainty in the covariance structure, including their dependence.

Marginal Sampling (marginal=TRUE):

Uses the marginal t-distribution of the mean:

$$\mu|Y, X, a_0 \sim t_{\nu^*-p+1}(\mu^*, \Lambda^*/(\kappa^*(\nu^* - p + 1)))$$

This is computationally faster and useful when you primarily care about inference on the mean vector, marginalizing over uncertainty in the covariance.

Value

If marginal=FALSE, a list with components:

`mu`	n_samples x p matrix of mean samples
`Sigma`	p x p x n_samples array of covariance samples

If marginal=TRUE, an n_samples x p matrix of mean samples.

Examples

```
library(MASS)
Sigma_true <- matrix(c(4, 1, 1, 2), 2, 2)
historical <- mvrnorm(50, mu = c(10, 5), Sigma = Sigma_true)
current <- mvrnorm(30, mu = c(10.5, 5.2), Sigma = Sigma_true)

pp <- powerprior_multivariate(historical, a0 = 0.5)
posterior <- posterior_multivariate(pp, current)

# Sample from joint distribution (covariance included)
samples_joint <- sample_posterior_multivariate(posterior, n_samples = 100)
cat("Mean samples shape:", dim(samples_joint$mu), "\n")
cat("Covariance samples shape:", dim(samples_joint$Sigma), "\n")

# Sample only means (faster)
samples_marginal <- sample_posterior_multivariate(posterior, n_samples = 100,
                                                   marginal = TRUE)
```

sample_posterior_univariate

Sample from Posterior Distribution (Univariate)

Description

Generates samples from the posterior distribution using the conjugate representation. Can sample the joint distribution (mu, sigma2) or just the marginal distribution of mu.

Usage

```
sample_posterior_univariate(posterior, n_samples = 1000, marginal = FALSE)
```

Arguments

`posterior`	Object of class "posterior_univariate" from posterior_univariate()
`n_samples`	Number of samples to generate (default: 1000)
`marginal`	Logical. If TRUE, samples only mu from t-distribution. If FALSE, samples joint (mu, sigma2) from NIX distribution (default: FALSE)

Value

If marginal=FALSE, a matrix with columns "mu" and "sigma2". If marginal=TRUE, a vector of mu samples.

Examples

```
# Generate data and compute posterior
historical <- rnorm(50, mean = 10, sd = 2)
current <- rnorm(30, mean = 10.5, sd = 2)
pp <- powerprior_univariate(historical, a0 = 0.5)
posterior <- posterior_univariate(pp, current)

# Sample from joint distribution
samples_joint <- sample_posterior_univariate(posterior, n_samples = 1000)

# Sample from marginal distribution of mu
samples_marginal <- sample_posterior_univariate(posterior, n_samples = 1000,
                                                marginal = TRUE)
```

Index