Package 'spmixW'

April 16, 2026

Title Bayesian Spatial Panel Data Models with Convex Combinations of Weight Matrices

Version 0.2.2

Description Bayesian Markov chain Monte Carlo (MCMC) estimation of spatial panel data models including Spatial Autoregressive (SAR), Spatial Durbin Model (SDM), Spatial Error Model (SEM), Spatial Durbin Error Model (SDEM), and Spatial Lag of X (SLX) specifications with fixed effects. Supports convex combinations of multiple spatial weight matrices and Bayesian Model Averaging (BMA) over subsets of weight matrices. Implements the convex combination spatial weight matrix methodology of Debarsy and LeSage (2021) <doi:10.1080/07350015.2020.1840993> and the Bayesian spatial panel data models of LeSage and Pace (2009, ISBN:9781420064247).

License GPL (>= 3)

Depends R (>= 4.1.0)

Imports Matrix, MASS, coda

Suggests testthat (>= 3.0.0), knitr, rmarkdown, spdep, sf, ggplot2, broom

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Mustapha Wasseja Mohammed [aut, cre]

Maintainer Mustapha Wasseja Mohammed <muswaseja@gmail.com>

Repository CRAN

Date/Publication 2026-04-16 09:20:02 UTC

Contents

`coef.spmixW` *Coefficient Extractor for spmixW Objects*

Description

Coefficient Extractor for spmixW Objects

Usage

```
## S3 method for class 'spmixW'
coef(object, ...)
```

Arguments

object	An object of class "spmixW".
...	Further arguments (ignored).

Value

Named numeric vector of posterior means.

`coef.spmixW_bma` *Coefficient Extractor for spmixW_bma Objects*

Description

Coefficient Extractor for spmixW_bma Objects

Usage

```
## S3 method for class 'spmixW_bma'
coef(object, ...)
```

Arguments

object	An object of class "spmixW_bma".
...	Further arguments (ignored).

Value

Named numeric vector of BMA-averaged posterior means.

compare_models *Compare Spatial Panel Model Specifications*

Description

Computes log-marginal likelihoods for SLX, SDM, and SDEM specifications and returns posterior model probabilities. A convenience wrapper around `lmarginal_panel`.

Usage

```
compare_models(formula, data, W, id, time = NULL, effects = "twoway", ...)
```

Arguments

`formula`	A formula of the form `y ~ x1 + x2 + ...`. An intercept is NOT included in the design matrix (fixed effects handle the level).
`data`	A data frame containing all variables referenced in the formula, plus the `id` and `time` columns.
`W`	A spatial weight matrix (N x N) or a list of M weight matrices for convex combination models. When a list is provided, the convex combination variant of the specified model is used automatically.
`id`	Character: name of the column in `data` identifying cross-sectional units (regions).
`time`	Character or NULL: name of the column identifying time periods. If `NULL`, a cross-sectional model (T=1) is estimated with `effects = "none"`.
`effects`	Character: fixed-effects specification. One of `"none"`, `"region"`, `"time"`, `"twoway"`. Default `"twoway"`.
`...`	Additional arguments passed to the prior list (e.g., `beta_mean`, `beta_var`, `nu`, `d0`).

Value

A data frame with columns: `model`, `log_marginal`, `probability`.

Examples

```
coords <- cbind(runif(50), runif(50))
W <- make_knw(coords, k = 5)
panel <- simulate_panel(N = 50, T = 8, W = W, rho = 0.4,
                        beta = c(1, -0.5), seed = 42)
comp <- compare_models(y ~ x1 + x2, data = panel, W = as.matrix(W),
                       id = "region", time = "year", effects = "twoway")
print(comp)
```

`demean_panel` *Panel Demeaning for Fixed Effects*

Description

Removes fixed effects from panel data via the within transformation. Data must be sorted by time then by spatial unit: all N regions in period 1, then all N regions in period 2, etc. (i.e., y_{it} is stored at position $i + (t-1) \times N$).

Usage

```
demean_panel(y, X, N, Time, model = 0L)
```

Arguments

`y`	Numeric vector of length NT: dependent variable.
`X`	Numeric matrix of dimension $NT \times k$: explanatory variables.
`N`	Integer: number of cross-sectional units (regions).
`Time`	Integer: number of time periods.
`model`	Integer controlling fixed-effects specification: **0** No demeaning (pooled). **1** Region (spatial) fixed effects only. **2** Time period fixed effects only. **3** Both region and time period fixed effects (two-way FE).

Details

The demeaning follows the standard panel FE within transformation:

- Model 1 (region FE): subtract region means across time.
- Model 2 (time FE): subtract time-period means across regions.
- Model 3 (two-way FE): subtract both region and time means, then add back the grand mean (to avoid double subtraction).

Value

A list with elements:

ywith Demeaned `y` (length NT).

xwith Demeaned `X` ($NT \times k$).

meanny Region means of y (length N); zero if `model %in% c(0,2)`.

meannx Region means of X ($N \times k$); zero if `model %in% c(0,2)`.

meanty Time means of y (length T); zero if `model %in% c(0,1)`.

meantx Time means of X ($T \times k$); zero if `model %in% c(0,1)`.

References

LeSage, J.P. and Pace, R.K. (2009). *Introduction to Spatial Econometrics*. Taylor & Francis/CRC Press.

Examples

```
N <- 10; Time <- 5; k <- 2
set.seed(42)
y <- rnorm(N * Time)
X <- matrix(rnorm(N * Time * k), ncol = k)
dm <- demean_panel(y, X, N, Time, model = 3)
# Verify the demeaned y has (approximately) zero region and time means
```

`eval_taylor_lndet` *Evaluate Taylor Series Log-Determinant at Given (rho, gamma)*

Description

Rapidly evaluates $\log|I - \rho W_c(\gamma)|$ using pre-computed trace terms from `log_det_taylor`.

Usage

```
eval_taylor_lndet(traces, rho, gamma, order = NULL)
```

Arguments

`traces`	Output from `log_det_taylor`.
`rho`	Numeric scalar: spatial autoregressive parameter.
`gamma`	Numeric vector of length M: convex weights (must sum to 1).
`order`	Integer: Taylor order to use (default: use all available). Must be <= `traces$max_order`.

Value

Scalar: approximate $\log|I - \rho W_c(\gamma)|$.

glance.spmixW *Glance Method for spmixW Objects*

Description

Returns a one-row data frame of model-level summary statistics.

Usage

```
glance.spmixW(x, ...)
```

Arguments

x	An object of class "spmixW".
...	Additional arguments (ignored).

Value

A one-row data frame.

glance.spmixW_bma *Glance Method for spmixW_bma Objects*

Description

Glance Method for spmixW_bma Objects

Usage

```
glance.spmixW_bma(x, ...)
```

Arguments

x	An object of class "spmixW_bma".
...	Additional arguments (ignored).

Value

A one-row data frame.

`lmarginal_panel` *Log-Marginal Likelihoods for Static Spatial Panel Models*

Description

Computes log-marginal likelihoods for three spatial panel specifications (SLX, SDM, SDEM) under diffuse priors on β, σ^2 and a uniform prior on ρ/λ. Used for Bayesian model comparison.

Usage

```
lmarginal_panel(y, X, W, N, Time, prior = list())
```

Arguments

`y`	Numeric vector of length NT (should be demeaned if FE are used).
`X`	Numeric matrix NT x k (WITHOUT intercept for FE models).
`W`	Spatial weight matrix (N x N).
`N`	Integer: number of cross-sectional units.
`Time`	Integer: number of time periods.
`prior`	Optional list with fields:
	lflag 0 = exact log-det, 1 = MC approximation (default).
	order MC order (default 50).
	iter MC iterations (default 30).
	rmin, rmax Bounds for rho grid.

Details

The SLX model has no spatial parameter, so its marginal is analytic. The SDM and SDEM marginals require numerical integration over ρ (or λ) using the trapezoid rule on a fine grid.

For SDM, the concentrated log-marginal profile is:

$$-(dof)\log(e_0'e_0 - 2\rho e_0'e_d + \rho^2 e_d'e_d) + T\log|I - \rho W|$$

where e_0, e_d are OLS residuals from y and Wy on the SDM design matrix.

Value

A list with:

logm_slx Log-marginal for SLX model.

logm_sdm Log-marginal for SDM model (integrated over rho).

logm_sdem Log-marginal for SDEM model (integrated over lambda).

lmarginal Vector of all three log-marginals.

probs Posterior model probabilities (assuming equal priors).

References

LeSage, J.P. (2014). "What Regional Scientists Need to Know about Spatial Econometrics." *Review of Regional Studies*, 44(1), 13-32.

LeSage, J.P. and Pace, R.K. (2009). *Introduction to Spatial Econometrics*. Taylor & Francis/CRC Press.

Examples

```
set.seed(1)
N <- 30; Time <- 10
coords <- cbind(runif(N), runif(N))
W <- as.matrix(normw(make_knw(coords, k = 5, row_normalise = FALSE)))
X <- matrix(rnorm(N * Time * 2), ncol = 2)
y <- rnorm(N * Time)
res <- lmarginal_panel(y, X, W, N, Time)
res$probs
```

---

logLik.spmixW *Log-Likelihood for spmixW Objects*

Description

Log-Likelihood for spmixW Objects

Usage

```
## S3 method for class 'spmixW'
logLik(object, ...)
```

Arguments

`object`	An object of class `"spmixW"`.
`...`	Further arguments (ignored).

Value

The log-likelihood evaluated at the posterior mean.

`log_det_exact` *Exact Log-Determinant via Eigenvalues*

Description

Computes $\log|I_N - \rho W|$ for a grid of ρ values using the eigenvalues of W. This is exact but requires computing all eigenvalues of W, which is $O(N^3)$.

Usage

```
log_det_exact(W, rmin = -1, rmax = 1, grid_step = 0.001)
```

Arguments

`W`	A square (sparse or dense) spatial weight matrix ($N \times N$).
`rmin`	Numeric scalar: lower bound of ρ grid (default `-1`).
`rmax`	Numeric scalar: upper bound of ρ grid (default `1`).
`grid_step`	Numeric scalar: grid spacing (default `0.001`).

Details

Given eigenvalues $\lambda_1, \ldots, \lambda_N$ of W:

$$\log|I_N - \rho W| = \sum_{i=1}^{N} \log(1 - \rho\lambda_i)$$

The result is pre-computed on a fine grid and used for griddy Gibbs sampling of the spatial autoregressive parameter.

Value

A matrix with two columns:

rho Grid of ρ values.

lndet Corresponding $\log|I_N - \rho W|$ values.

References

LeSage, J.P. and Pace, R.K. (2009). *Introduction to Spatial Econometrics*. Taylor & Francis/CRC Press.

Examples

```
W <- matrix(c(0, 0.5, 0.5, 0,
              0.5, 0, 0, 0.5,
              0.5, 0, 0, 0.5,
              0, 0.5, 0.5, 0), 4, 4)
detval <- log_det_exact(W)
head(detval)
```

---

log_det_mc *Monte Carlo Log-Determinant Approximation (Pace-Barry 1999)*

Description

Approximates $\log|I_N - \rho W|$ using the stochastic trace estimator of Barry and Pace (1999). Suitable for large spatial weight matrices where exact eigenvalue computation is infeasible.

Usage

```
log_det_mc(W, rmin = -1, rmax = 1, grid_step = 0.001, order = 50L, iter = 30L)
```

Arguments

`W`	A square sparse spatial weight matrix ($N \times N$).
`rmin`	Numeric scalar: lower bound of ρ grid (default `-1`).
`rmax`	Numeric scalar: upper bound of ρ grid (default `1`).
`grid_step`	Numeric scalar: grid spacing (default `0.001`).
`order`	Integer: number of terms in the Taylor expansion (default `50`).
`iter`	Integer: number of random vectors for trace estimation (default `30`).

Details

Uses the identity:

$$\log|I - \rho W| = -\sum_{j=1}^{\infty} \frac{\rho^j}{j} \text{tr}(W^j)$$

The traces $\text{tr}(W^j)$ are estimated stochastically:

$$\text{tr}(W^j) \approx \frac{1}{q} \sum_{l=1}^{q} u_l' W^j u_l$$

where u_l are random $N(0, I)$ vectors.

The second-order trace $\text{tr}(W^2)$ is computed exactly as $\text{tr}(W'W) = \sum_{ij} w_{ij}^2$ for greater accuracy.

Value

A matrix with two columns:

rho Grid of ρ values.

lndet Corresponding approximate $\log|I_N - \rho W|$ values.

References

Pace, R.K. and Barry, J.P. (1997). "Quick Computation of Spatial Autoregressive Estimators." *Geographical Analysis*, 29(3), 232-247.

Barry, R.P. and Pace, R.K. (1999). "Monte Carlo estimates of the log determinant of large sparse matrices." *Linear Algebra and its Applications*, 289, 41-54.

Examples

```
library(Matrix)
N <- 100
W <- sparseMatrix(
  i = c(1:N, 1:(N-1)),
  j = c(c(2:N, 1), 2:N),
  x = rep(0.5, 2*N - 1),
  dims = c(N, N)
)
detval <- log_det_mc(W)
head(detval)
```

---

log_det_taylor *Taylor Series Log-Determinant for Convex Combination of Weight Matrices*

Description

Pre-computes trace cross-terms for the Taylor series approximation to $\log|I - \rho W_c(\gamma)|$ where $W_c = \sum_{m=1}^{M} \gamma_m W_m$.

Usage

```
log_det_taylor(Wlist, max_order = 4L)
```

Arguments

`Wlist`	A list of M sparse or dense weight matrices, each $N \times N$.
`max_order`	Integer: maximum Taylor order (default 4, supports 2-8). Higher orders give more accurate log-det approximations at the cost of more pre-computation time. The number of trace terms at order p is M^p, so for M=3 and order 8 this is 6561 terms.

Details

The Taylor approximation is:

$$\log|I - \rho W_c| \approx -\sum_{p=2}^{\text{order}} \frac{\rho^p}{p} \text{tr}(W_c^p)$$

(the p=1 term vanishes for zero-diagonal W).

Since $W_c = \sum \gamma_m W_m$, the trace expands via multinomial:

$$\text{tr}(W_c^p) = \sum_{i_1,\ldots,i_p} \gamma_{i_1} \cdots \gamma_{i_p} \text{tr}(W_{i_1} \cdots W_{i_p})$$

These cross-terms are pre-computed once (expensive for high orders) and then rapidly evaluated for any (ρ, γ) via Kronecker products.

For M=3 weight matrices, the number of trace terms by order:

Order	Terms
4	81
5	243
6	729
7	2187
8	6561

Value

A list with elements:

traces A named list where `traces[[p]]` is a vector of length M^p containing all $\text{tr}(W_{i_1} \cdots W_{i_p})$ cross-terms for order p (p = 2, ..., max_order).

max_order The maximum Taylor order stored.

M Number of weight matrices.

N Dimension of each weight matrix.

References

Debarsy, N. and LeSage, J.P. (2021). "Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices." *Journal of Business & Economic Statistics*, 40(2), 547-558.

Examples

```
N <- 20
W1 <- matrix(0, N, N); W2 <- matrix(0, N, N)
for (i in 1:N) { W1[i, (i %% N) + 1] <- 1; W2[i, ((i-2) %% N) + 1] <- 1 }
W1 <- W1 / rowSums(W1); W2 <- W2 / rowSums(W2)
traces <- log_det_taylor(list(W1, W2), max_order = 6)
eval_taylor_lndet(traces, rho = 0.5, gamma = c(0.7, 0.3))
```

make_knw *Create k-Nearest-Neighbours Spatial Weight Matrix*

Description

Constructs a row-normalised binary spatial weight matrix based on k-nearest neighbours from a coordinate matrix.

Usage

```
make_knw(coords, k, row_normalise = TRUE)
```

Arguments

coords	An $N \times 2$ matrix of coordinates (e.g., longitude/latitude or projected x/y).
k	Integer: number of nearest neighbours.
row_normalise	Logical: if TRUE (default), row-normalise the resulting matrix.

Details

Euclidean distances are computed between all pairs of coordinates. For each unit, the k closest neighbours receive weight 1 (before normalisation). The matrix is generally asymmetric (i may be j's neighbour without j being i's neighbour).

Value

A sparse $N \times N$ weight matrix. If row_normalise = TRUE, each row sums to 1.

Examples

```
set.seed(1)
coords <- cbind(runif(20), runif(20))
W <- make_knw(coords, k = 5)
dim(W)
range(Matrix::rowSums(W))  # all 1 if row-normalised
```

`model_probs` *Posterior Model Probabilities from Log-Marginal Likelihoods*

Description

Converts a vector of log-marginal likelihoods to posterior model probabilities assuming equal prior model probabilities.

Usage

```
model_probs(lmarginal)
```

Arguments

`lmarginal`	Numeric vector of log-marginal likelihood values, one per model.

Details

Assumes equal prior probabilities across models. Computes:

$$p(M_j|y) = \frac{\exp(\ell_j - \max(\ell))}{\sum_i \exp(\ell_i - \max(\ell))}$$

where ℓ_j is the log-marginal likelihood for model j. The subtraction of the maximum prevents numerical overflow.

Value

A numeric vector of posterior model probabilities summing to 1.

References

LeSage, J.P. (2014). "What Regional Scientists Need to Know about Spatial Econometrics." *Review of Regional Studies*, 44(1), 13-32.

Examples

```
lm <- c(-100, -98, -105)
model_probs(lm)
```

normw *Row-Normalise a Spatial Weight Matrix*

Description

Divides each row of a spatial weight matrix by its row sum so that all rows sum to one. Zero-sum rows (isolates) are left unchanged.

Usage

```
normw(W)
```

Arguments

W	A square matrix or sparse matrix (class dgCMatrix or similar).

Value

A sparse matrix of the same dimension with rows summing to 1 (except for isolate rows that remain zero).

Examples

```
W <- matrix(c(0, 1, 1, 0,
              1, 0, 0, 1,
              1, 0, 0, 1,
              0, 1, 1, 0), 4, 4)
Wn <- normw(W)
Matrix::rowSums(Wn)  # all ones
```

ols_panel *Bayesian OLS Panel Model with Fixed Effects*

Description

Estimates a Bayesian panel regression model (no spatial component) with optional spatial and/or time-period fixed effects via MCMC sampling. Supports both homoscedastic and heteroscedastic error structures.

Usage

```
ols_panel(y, X, N, Time, ndraw = 5500L, nomit = 1500L, prior = list())
```

Arguments

`y`	Numeric vector of length NT: dependent variable. Data must be sorted by time then by spatial unit: all N regions in period 1, then all N in period 2, etc.
`X`	Numeric matrix of dimension $NT \times k$: explanatory variables.
`N`	Integer: number of cross-sectional units.
`Time`	Integer: number of time periods.
`ndraw`	Integer: total number of MCMC draws (including burn-in).
`nomit`	Integer: number of initial draws to discard as burn-in.
`prior`	A list of prior hyperparameters:

model Integer 0-3: fixed-effects specification (0 = pooled, 1 = region FE, 2 = time FE, 3 = both). Default 0.

rval Numeric > 0: degrees-of-freedom for heteroscedastic errors. Each $v_i \sim \chi^2(r)/r$. Set to 0 for homoscedastic. Default 4.

beta_mean Numeric vector of length k: prior mean for β. Default `rep(0, k)`.

beta_var Numeric $k \times k$ matrix: prior variance for β. Default `diag(k) * 1e12` (diffuse).

nu Numeric >= 0: shape parameter for inverse-gamma prior on σ^2. Default 0 (diffuse).

d0 Numeric >= 0: scale parameter for inverse-gamma prior on σ^2. Default 0 (diffuse).

thin Integer >= 1: thinning interval. Default 1 (no thinning).

Details

The model is:

$$y = X\beta + \iota_T \otimes \mu + \nu \otimes \iota_N + \varepsilon$$

where $\varepsilon \sim N(0, \sigma^2 V)$, $V = \text{diag}(v_1, \ldots, v_{NT})$.

MCMC sampling steps:

1. Draw $\beta|\sigma^2, V, y$ from the multivariate normal posterior (conjugate update).
2. Draw $\sigma^2|\beta, V, y$ from the inverse-gamma posterior.
3. (Heteroscedastic only) Draw each $v_i|\beta, \sigma^2, y$ from $(\epsilon_i^2/\sigma^2 + r)/\chi^2(r+1)$.

Value

An S3 object of class `"spmixW"` containing:

beta Posterior mean of β (length k).

sige Posterior mean of σ^2.

bdraw `coda::mcmc` object: $(ndraw - nomit) \times k$ matrix of retained β draws.

sdraw `coda::mcmc` object: retained σ^2 draws.

vmean Posterior mean of the variance weights v_i (length NT; all ones if homoscedastic).

yhat Fitted values including fixed effects.

resid Residuals $y - \hat{y}$.

rsqr R-squared.

corr2 Squared correlation between actual and fitted demeaned values.

sfe Spatial fixed effects (if `model %in% c(1,3)`).

tfe Time fixed effects (if `model %in% c(2,3)`).

intercept Estimated intercept.

tstat Posterior t-statistics (mean / sd of draws).

nobs, nvar, N, Time, model, meth Metadata.

time Wall-clock time in seconds.

References

LeSage, J.P. and Pace, R.K. (2009). *Introduction to Spatial Econometrics*. Taylor & Francis/CRC Press.

Geweke, J. (1993). "Bayesian Treatment of the Independent Student-t Linear Model." *Journal of Applied Econometrics*, 8(S1), S19-S40.

Examples

```
set.seed(123)
N <- 20; Time <- 10; k <- 2
beta_true <- c(1.5, -0.8)
X <- matrix(rnorm(N * Time * k), ncol = k)
y <- X %*% beta_true + rnorm(N * Time, sd = 0.5)
res <- ols_panel(y, X, N, Time, ndraw = 2000, nomit = 500,
                 prior = list(model = 1, rval = 4))
print(res)
```

---

plot.spmixW *Plot Method for spmixW Objects*

Description

Produces a 2x2 diagnostic plot layout: trace and density for the spatial parameter, trace for sigma^2, and an effects comparison (or gamma densities for convex combination models).

Usage

```
## S3 method for class 'spmixW'
plot(x, ...)
```

Arguments

`x`	An object of class `"spmixW"`.
`...`	Further arguments (ignored).

Value

Invisible NULL.

`plot.spmixW_bma` *Plot Method for spmixW_bma Objects*

Description

Produces a multi-panel plot showing model probabilities and BMA posterior densities.

Usage

```
## S3 method for class 'spmixW_bma'
plot(x, ...)
```

Arguments

`x`	An object of class `"spmixW_bma"`.
`...`	Further arguments (ignored).

Value

Invisible NULL.

`print.spmixW` *Print Method for spmixW Objects*

Description

Print Method for spmixW Objects

Usage

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

Arguments

`x`	An object of class `"spmixW"`.
`digits`	Integer: number of significant digits to print.
`...`	Further arguments (ignored).

Value

Invisibly returns `x`.

print.spmixW_bma *Print Method for spmixW_bma Objects*

Description

Print Method for spmixW_bma Objects

Usage

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

Arguments

x	An object of class "spmixW_bma".
digits	Integer: significant digits.
...	Further arguments (ignored).

Value

Invisibly returns x.

sar_conv_bma *Bayesian Model Averaging for SAR Convex Combination Panel*

Description

Estimates SAR convex combination models for all $2^M - 1$ non-empty subsets of M weight matrices and averages estimates by posterior model probabilities computed from log-marginal likelihoods.

Usage

```
sar_conv_bma(
  y,
  X,
  Wlist,
  N,
  Time,
  ndraw = 25000L,
  nomit = 5000L,
  prior = list()
)
```

Arguments

y	Numeric vector of length NT.
X	Numeric matrix NT x k.
Wlist	A list of M spatial weight matrices (each N x N).
N	Integer: number of cross-sectional units.
Time	Integer: number of time periods.
ndraw	Integer: MCMC draws per model (recommend >= 10000).
nomit	Integer: burn-in draws per model.
prior	List of prior hyperparameters (passed to each model).

Value

An S3 object of class "spmixW_bma" containing:

beta BMA posterior mean of beta.

rho BMA posterior mean of rho.

gamma BMA posterior mean of gamma (M x 1).

sige BMA posterior mean of sigma^2.

probs Vector of posterior model probabilities.

logm Vector of log-marginal likelihoods per model.

bdraw, pdraw, sdraw, gdraw BMA-averaged MCMC draws.

direct, indirect, total BMA-averaged effects draws.

subsets Matrix showing which W's are in each model.

nmodels Number of models evaluated.

per_model List of per-model results summaries.

References

Debarsy, N. and LeSage, J.P. (2021). "Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices." *Journal of Business & Economic Statistics*, 40(2), 547-558.

---

sar_conv_panel *Bayesian SAR Panel with Convex Combination of Weight Matrices*

Description

Estimates a SAR panel model where the spatial weight matrix is a convex combination $W_c = \sum_{m=1}^{M} \gamma_m W_m$ with γ on the unit simplex. Uses Metropolis-Hastings for both ρ and γ, with Taylor series log-determinant approximation.

Usage

```
sar_conv_panel(
  y,
  X,
  Wlist,
  N,
  Time,
  ndraw = 25000L,
  nomit = 5000L,
  prior = list()
)
```

Arguments

y	Numeric vector of length NT.
X	Numeric matrix NT x k.
Wlist	A list of M spatial weight matrices (each N x N).
N	Integer: number of cross-sectional units.
Time	Integer: number of time periods.
ndraw	Integer: total MCMC draws (recommend >= 10000).
nomit	Integer: burn-in draws.
prior	List of prior hyperparameters. See Details.

Details

The model is:

$$y = \rho W_c(\gamma) y + X\beta + \text{FE} + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_{NT})$$

The MCMC sampler draws:

1. β|rest from multivariate normal (conjugate).
2. σ^2|rest from inverse-gamma.
3. ρ|rest via Metropolis-Hastings random walk with adaptive tuning (target acceptance 40-60\
4. γ|rest via Metropolis-Hastings with reversible jump proposal on the simplex (matching LeSage's coin-flip method).

Pre-computes $\tilde{y} = [y, -W_1 y, \ldots, -W_M y]$ so that for any (ρ, γ): $(I - \rho W_c)y = \tilde{y}\omega$ where $\omega = (1, \rho\gamma_1, \ldots, \rho\gamma_M)'$.

Value

An S3 object of class "spmixW" with:

beta Posterior mean of beta.

rho Posterior mean of rho.

gamma Posterior mean of gamma (M x 1).

bdraw, pdraw, sdraw, gdraw MCMC draws for beta, rho, sigma, gamma.

rho_acc_rate Acceptance rate for rho MH sampler.

gamma_acc_rate Acceptance rate for gamma MH sampler.

direct, indirect, total Effects estimates.

traces Pre-computed Taylor traces (for reuse).

Taylor approximation accuracy

The convex combination models use a Taylor series approximation for the log-determinant. The default order is 6 (extending the original order 4 of Debarsy and LeSage 2021). Approximation accuracy improves with larger N: at N=500 and rho=0.6, expect approximately 0.13 upward bias in rho; at N=3000+ the bias is negligible (<0.02). Users can control this via `prior$taylor_order`. For small-N applications where precise rho estimation is critical, compare results against `sar_panel` with a known single W matrix as a benchmark.

References

Debarsy, N. and LeSage, J.P. (2021). "Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices." *Journal of Business & Economic Statistics*, 40(2), 547-558.

LeSage, J.P. (2020). "Fast MCMC estimation of multiple W-matrix spatial regression models and Metropolis-Hastings Monte Carlo log-marginal likelihoods." *Journal of Geographical Systems*, 22(1), 47-75.

`sar_panel` *Bayesian SAR Panel Model with Fixed Effects*

Description

Estimates a Bayesian Spatial Autoregressive (SAR) panel model via MCMC:

$$y = \rho W y + X\beta + \text{FE} + \varepsilon$$

with griddy Gibbs sampling for ρ and LeSage-Pace scalar effects estimates (direct, indirect, total).

Usage

```
sar_panel(y, X, W, N, Time, ndraw = 5500L, nomit = 1500L, prior = list())
```

Arguments

`y`	Numeric vector of length NT.
`X`	Numeric matrix $NT \times k$.
`W`	Spatial weight matrix ($N \times N$ or $NT \times NT$).
`N`	Integer: number of cross-sectional units.

`Time`	Integer: number of time periods.
`ndraw`	Integer: total MCMC draws (including burn-in).
`nomit`	Integer: burn-in draws to discard.
`prior`	List of prior hyperparameters (see `ols_panel` for common fields). Additional fields:
	lflag 0 = exact log-det, 1 = MC approximation (default).
	order Taylor order for MC log-det (default 50).
	iter MC iterations for log-det (default 30).
	rmin, rmax Bounds for ρ grid (default `c(-1,1)`).
	lndet Pre-computed log-det grid (2-column matrix) to reuse.

Details

The griddy Gibbs sampler for ρ evaluates the conditional log-posterior on a fine grid using the concentrated likelihood (beta integrated out), then uses inverse-CDF sampling via the trapezoid rule.

Effects are computed using the scalar summary measures of LeSage and Pace (2009, Ch. 4) based on stochastic trace estimates of powers of W.

Value

An S3 object of class `"spmixW"` with all fields from `ols_panel` plus:

rho Posterior mean of ρ.

pdraw `coda::mcmc` of ρ draws.

direct Matrix (p x 5): direct effects (mean, t-stat, p-value, lower, upper).

indirect Matrix (p x 5): indirect effects.

total Matrix (p x 5): total effects.

direct_draws, indirect_draws, total_draws MCMC draws of effects.

lndet Log-determinant grid (for reuse in subsequent calls).

References

LeSage, J.P. and Pace, R.K. (2009). *Introduction to Spatial Econometrics*. Taylor & Francis/CRC Press.

Examples

```
set.seed(1)
N <- 30; Time <- 10; rho_true <- 0.5
coords <- cbind(runif(N), runif(N))
W <- normw(make_knw(coords, k = 5, row_normalise = FALSE))
W <- as.matrix(W)
Wbig <- kronecker(diag(Time), W)
X <- matrix(rnorm(N * Time * 2), ncol = 2)
y <- solve(diag(N*Time) - rho_true * Wbig) %*% (X %*% c(1, -0.5) + rnorm(N*Time))
```

```
res <- sar_panel(as.numeric(y), X, W, N, Time, ndraw = 5000, nomit = 2000,
                 prior = list(model = 0, rval = 0))
print(res)
```

---

sdem_conv_bma *Bayesian Model Averaging for SDEM Convex Combination Panel*

Description

Bayesian Model Averaging for SDEM Convex Combination Panel

Usage

```
sdem_conv_bma(
  y,
  X,
  Wlist,
  N,
  Time,
  ndraw = 25000L,
  nomit = 5000L,
  prior = list()
)
```

Arguments

y	Numeric vector of length NT.
X	Numeric matrix NT x k.
Wlist	A list of M spatial weight matrices (each N x N).
N	Integer: number of cross-sectional units.
Time	Integer: number of time periods.
ndraw	Integer: MCMC draws per model (recommend >= 10000).
nomit	Integer: burn-in draws per model.
prior	List of prior hyperparameters (passed to each model).

Value

An S3 object of class "spmixW_bma".

`sdem_conv_panel` *Bayesian SDEM Panel with Convex Combination of Weight Matrices*

Description

Estimates a SDEM panel model with $W_c = \sum \gamma_m W_m$:

$$y = X\beta + W_1 X\theta_1 + \ldots + W_M X\theta_M + u, \quad u = \lambda W_c u + \varepsilon$$

Usage

```
sdem_conv_panel(
  y,
  X,
  Wlist,
  N,
  Time,
  ndraw = 25000L,
  nomit = 5000L,
  prior = list()
)
```

Arguments

`y`	Numeric vector of length NT.
`X`	Numeric matrix NT x k.
`Wlist`	A list of M spatial weight matrices (each N x N).
`N`	Integer: number of cross-sectional units.
`Time`	Integer: number of time periods.
`ndraw`	Integer: total MCMC draws (recommend >= 10000).
`nomit`	Integer: burn-in draws.
`prior`	List of prior hyperparameters. See Details.

Details

Augments X with `[X, W1*X, W2*X, ..., WM*X]` and calls `sem_conv_panel`.

For SDEM, direct effects are the β coefficients, indirect effects are the θ_m coefficients summed across W matrices, and total = direct + indirect. The spatial error parameter does not affect effects.

Value

An S3 object of class `"spmixW"` with convex combination fields plus SDEM-specific effects.

Taylor approximation accuracy

See `sar_conv_panel` for details on Taylor order and accuracy.

References

Debarsy, N. and LeSage, J.P. (2021). "Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices." *Journal of Business & Economic Statistics*, 40(2), 547-558.

`sdem_panel` *Bayesian SDEM Panel Model with Fixed Effects*

Description

Estimates a Spatial Durbin Error Model (SDEM) panel via MCMC. This is a thin wrapper around `sem_panel` that augments X with WX.

Usage

```
sdem_panel(y, X, W, N, Time, ndraw = 5500L, nomit = 1500L, prior = list())
```

Arguments

`y`	Numeric vector of length NT.
`X`	Numeric matrix $NT \times k$.
`W`	Spatial weight matrix ($N \times N$ or $NT \times NT$).
`N`	Integer: number of cross-sectional units.
`Time`	Integer: number of time periods.
`ndraw`	Integer: total MCMC draws (including burn-in).
`nomit`	Integer: burn-in draws to discard.
`prior`	List of prior hyperparameters (see `ols_panel` for common fields). Additional fields: **lflag** 0 = exact log-det, 1 = MC approximation (default). **order** Taylor order for MC log-det (default 50). **iter** MC iterations for log-det (default 30). **rmin, rmax** Bounds for ρ grid (default `c(-1,1)`). **ldet** Pre-computed log-det grid (2-column matrix) to reuse.

Details

$$y = X\beta + WX\theta + \text{FE} + u, \quad u = \lambda W u + \varepsilon$$

For SDEM, effects decomposition is straightforward (no spatial multiplier on X, unlike SDM): direct effects are β, indirect effects are θ, and total = $\beta + \theta$. The spatial error parameter λ does not affect the effects decomposition.

Value

An S3 object of class "spmixW" with SEM fields plus SDEM-specific effects:

direct Direct effects = coefficients on X (β).

indirect Indirect effects = coefficients on WX (θ).

total Total = direct + indirect.

References

Debarsy, N. and LeSage, J.P. (2021). "Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices." *Journal of Business & Economic Statistics*, 40(2), 547-558.

Examples

```
set.seed(1)
N <- 30; Time <- 10; lambda_true <- 0.5
coords <- cbind(runif(N), runif(N))
W <- as.matrix(normw(make_knw(coords, k = 5, row_normalise = FALSE)))
Wbig <- kronecker(diag(Time), W)
X <- matrix(rnorm(N * Time * 2), ncol = 2)
WX <- Wbig %*% X
u <- solve(diag(N*Time) - lambda_true * Wbig) %*% rnorm(N*Time)
y <- X %*% c(1, -0.5) + WX %*% c(0.3, 0.2) + u
res <- sdem_panel(as.numeric(y), X, W, N, Time, ndraw = 5000, nomit = 2000)
print(res)
```

---

sdm_conv_bma *Bayesian Model Averaging for SDM Convex Combination Panel*

Description

Bayesian Model Averaging for SDM Convex Combination Panel

Usage

```
sdm_conv_bma(
  y,
  X,
  Wlist,
  N,
  Time,
  ndraw = 25000L,
  nomit = 5000L,
  prior = list()
)
```

Arguments

y	Numeric vector of length NT.
X	Numeric matrix NT x k.
Wlist	A list of M spatial weight matrices (each N x N).
N	Integer: number of cross-sectional units.
Time	Integer: number of time periods.
ndraw	Integer: MCMC draws per model (recommend >= 10000).
nomit	Integer: burn-in draws per model.
prior	List of prior hyperparameters (passed to each model).

Value

An S3 object of class "spmixW_bma".

---

sdm_conv_panel *Bayesian SDM Panel with Convex Combination of Weight Matrices*

Description

Estimates a SDM panel model with $W_c = \sum \gamma_m W_m$:

$$y = \rho W_c y + X\beta + W_1 X\theta_1 + \ldots + W_M X\theta_M + \varepsilon$$

Usage

```
sdm_conv_panel(
  y,
  X,
  Wlist,
  N,
  Time,
  ndraw = 25000L,
  nomit = 5000L,
  prior = list()
)
```

Arguments

y	Numeric vector of length NT.
X	Numeric matrix NT x k.
Wlist	A list of M spatial weight matrices (each N x N).
N	Integer: number of cross-sectional units.
Time	Integer: number of time periods.
ndraw	Integer: total MCMC draws (recommend >= 10000).
nomit	Integer: burn-in draws.
prior	List of prior hyperparameters. See Details.

Details

Unlike the standard SDM wrapper, the WX terms are pre-computed for each individual W_m since the convex combination W_c changes at each MCMC iteration. The augmented X is `[X, W1*X, W2*X, ..., WM*X]`.

Internally augments X with `[X, W1*X, W2*X, ..., WM*X]` and then calls `sar_conv_panel` for estimation. The effects decomposition accounts for all M sets of WX coefficients.

Value

An S3 object of class `"spmixW"` with convex combination fields.

Taylor approximation accuracy

See `sar_conv_panel` for details on Taylor order and accuracy.

References

Debarsy, N. and LeSage, J.P. (2021). "Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices." *Journal of Business & Economic Statistics*, 40(2), 547-558.

`sdm_panel` *Bayesian SDM Panel Model with Fixed Effects*

Description

Estimates a Spatial Durbin Model (SDM) panel via MCMC. This is a thin wrapper around `sar_panel` that augments the X matrix with WX before calling the SAR estimator, then computes SDM-specific effects.

Usage

```
sdm_panel(y, X, W, N, Time, ndraw = 5500L, nomit = 1500L, prior = list())
```

Arguments

`y`	Numeric vector of length NT.
`X`	Numeric matrix $NT \times k$.
`W`	Spatial weight matrix ($N \times N$ or $NT \times NT$).
`N`	Integer: number of cross-sectional units.
`Time`	Integer: number of time periods.
`ndraw`	Integer: total MCMC draws (including burn-in).
`nomit`	Integer: burn-in draws to discard.
`prior`	List of prior hyperparameters (see `ols_panel` for common fields). Additional fields:

lflag 0 = exact log-det, 1 = MC approximation (default).

order Taylor order for MC log-det (default 50).

iter MC iterations for log-det (default 30).

rmin, rmax Bounds for ρ grid (default `c(-1,1)`).

lndet Pre-computed log-det grid (2-column matrix) to reuse.

Details

$$y = \rho W y + X\beta + WX\theta + \text{FE} + \varepsilon$$

Internally, this function augments X with WX and calls `sar_panel`, following the delegation pattern of LeSage and Pace (2009, Ch. 10). The SDM effects computation differs from SAR because the spatial multiplier $(I - \rho W)^{-1}$ acts on both βI and θW, yielding different direct/indirect decompositions.

Value

An S3 object of class `"spmixW"` with all SAR fields. The `beta` vector contains $[\beta', \theta']'$ (length $2k$ or $2k - 1$ if X has an intercept). Effects are SDM-style: both β and θ contribute to the spatial multiplier.

References

LeSage, J.P. and Pace, R.K. (2009). *Introduction to Spatial Econometrics*. Taylor & Francis/CRC Press.

Examples

```
set.seed(1)
N <- 30; Time <- 10; rho_true <- 0.4
coords <- cbind(runif(N), runif(N))
W <- as.matrix(normw(make_knw(coords, k = 5, row_normalise = FALSE)))
Wbig <- kronecker(diag(Time), W)
X <- matrix(rnorm(N * Time * 2), ncol = 2)
WX <- Wbig %*% X
y <- solve(diag(N*Time) - rho_true * Wbig) %*%
       (X %*% c(1, -0.5) + WX %*% c(0.3, 0.2) + rnorm(N*Time))
res <- sdm_panel(as.numeric(y), X, W, N, Time, ndraw = 5000, nomit = 2000)
print(res)
```

`sem_conv_panel` *Bayesian SEM Panel with Convex Combination of Weight Matrices*

Description

Estimates a SEM panel model with $W_c = \sum \gamma_m W_m$:

$$y = X\beta + u, \quad u = \lambda W_c u + \varepsilon$$

Usage

```
sem_conv_panel(
  y,
  X,
  Wlist,
  N,
  Time,
  ndraw = 25000L,
  nomit = 5000L,
  prior = list()
)
```

Arguments

`y`	Numeric vector of length NT.
`X`	Numeric matrix NT x k.
`Wlist`	A list of M spatial weight matrices (each N x N).
`N`	Integer: number of cross-sectional units.
`Time`	Integer: number of time periods.
`ndraw`	Integer: total MCMC draws (recommend >= 10000).
`nomit`	Integer: burn-in draws.
`prior`	List of prior hyperparameters. See Details.

Details

Follows the same MH structure as `sar_conv_panel` but the spatial parameter enters the error structure. The pre-computed arrays `ys` and `xs` represent $(I - \lambda W_c)$ filtering.

Value

An S3 object of class `"spmixW"` with gamma, rho (lambda), acceptance rates, and MCMC draws.

Taylor approximation accuracy

See `sar_conv_panel` for details on Taylor order and accuracy.

References

Debarsy, N. and LeSage, J.P. (2021). "Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices." *Journal of Business & Economic Statistics*, 40(2), 547-558.

`sem_panel` *Bayesian SEM Panel Model with Fixed Effects*

Description

Estimates a Bayesian Spatial Error Model (SEM) panel via MCMC:

$$y = X\beta + \text{FE} + u, \quad u = \lambda W u + \varepsilon$$

with griddy Gibbs sampling for λ.

Usage

```
sem_panel(y, X, W, N, Time, ndraw = 5500L, nomit = 1500L, prior = list())
```

Arguments

`y`	Numeric vector of length NT.
`X`	Numeric matrix $NT \times k$.
`W`	Spatial weight matrix ($N \times N$ or $NT \times NT$).
`N`	Integer: number of cross-sectional units.
`Time`	Integer: number of time periods.
`ndraw`	Integer: total MCMC draws (including burn-in).
`nomit`	Integer: burn-in draws to discard.
`prior`	List of prior hyperparameters (see `ols_panel` for common fields). Additional fields: **lflag** 0 = exact log-det, 1 = MC approximation (default). **order** Taylor order for MC log-det (default 50). **iter** MC iterations for log-det (default 30). **rmin, rmax** Bounds for ρ grid (default `c(-1,1)`). **lndet** Pre-computed log-det grid (2-column matrix) to reuse.

Details

The SEM differs from SAR in that the spatial parameter enters the error structure. The MCMC sampler filters both y and X by $(I - \lambda W)$ before drawing β. The griddy Gibbs for λ evaluates the concentrated log-posterior on a grid, where for each grid point the filtered data $(I - \lambda W)y$ and $(I - \lambda W)X$ are used to compute the residual sum of squares.

Value

An S3 object of class `"spmixW"` with fields similar to `ols_panel` plus:

rho Posterior mean of λ (spatial error parameter).

pdraw `coda::mcmc` of λ draws.

lndet Log-determinant grid.

References

LeSage, J.P. and Pace, R.K. (2009). *Introduction to Spatial Econometrics*. Taylor & Francis/CRC Press.

Examples

```
set.seed(1)
N <- 30; Time <- 10; lambda_true <- 0.5
coords <- cbind(runif(N), runif(N))
W <- as.matrix(normw(make_knw(coords, k = 5, row_normalise = FALSE)))
Wbig <- kronecker(diag(Time), W)
X <- matrix(rnorm(N * Time * 2), ncol = 2)
u <- solve(diag(N*Time) - lambda_true * Wbig) %*% rnorm(N*Time)
y <- X %*% c(1, -0.5) + u
res <- sem_panel(as.numeric(y), X, W, N, Time, ndraw = 5000, nomit = 2000,
                 prior = list(model = 0, rval = 0))
print(res)
```

---

simulate_panel *Simulate Spatial Panel Data*

Description

Generates synthetic spatial panel data from a SAR/SDM/SEM/SDEM DGP, returning a ready-to-use data frame for `spmodel`.

Usage

```
simulate_panel(
  N,
  Time = 10L,
  W,
  gamma = NULL,
  rho = 0,
  beta,
  theta = NULL,
  sigma2 = 1,
  effects = "twoway",
```

```
  seed = NULL
)
```

Arguments

N	Integer: number of cross-sectional units (regions).
Time	Integer: number of time periods (use 1 for cross-sectional).
W	A single N x N weight matrix, or a list of M weight matrices for convex combination DGPs.
gamma	Numeric vector of convex weights (required when W is a list, must sum to 1). Ignored when W is a single matrix.
rho	Numeric: spatial autoregressive parameter (default 0).
beta	Numeric vector of true regression coefficients (length k).
theta	Numeric vector of WX coefficients for SDM/SDEM DGPs (default NULL). Must have same length as beta if provided.
sigma2	Numeric: error variance (default 1).
effects	Character: fixed-effects specification. "none", "region", "time", or "twoway" (default "twoway").
seed	Integer or NULL: random seed for reproducibility.

Details

The DGP is:

$$y = (I_{NT} - \rho W_c)^{-1}(X\beta + W_c X\theta + \text{FE} + \varepsilon)$$

where $W_c = \sum \gamma_m W_m$ (or just W if a single matrix), θ is omitted for SAR/SEM DGPs, and FE are generated as $\mu_i = i/N$ (region) and $\nu_t = t/T$ (time).

Value

A data frame with columns:

region Integer region identifier (1 to N).

year Integer time period identifier (omitted if Time = 1).

y Simulated response variable.

x1, x2, ... Simulated predictor variables (standard normal).

The data frame is sorted by time then region, ready for spmodel.

Examples

```
coords <- cbind(runif(80), runif(80))
W1 <- make_knw(coords, k = 4)
W2 <- make_knw(coords, k = 8)

panel <- simulate_panel(
  N = 80, T = 10,
```

```
  W = list(W1, W2),
  gamma = c(0.7, 0.3),
  rho = 0.5,
  beta = c(1, -1),
  seed = 42
)

res <- spmodel(y ~ x1 + x2, data = panel,
               W = list(geography = W1, trade = W2),
               model = "sar",
               id = "region", time = "year",
               effects = "twoway",
               ndraw = 8000, nomit = 2000)
print(res)
```

`slx_panel` *Bayesian SLX Panel Model with Fixed Effects*

Description

Estimates a Spatial Lag of X (SLX) panel model via MCMC. This is a thin wrapper around `ols_panel` that augments X with WX.

Usage

```
slx_panel(y, X, W, N, Time, ndraw = 5500L, nomit = 1500L, prior = list())
```

Arguments

`y`	Numeric vector of length NT.
`X`	Numeric matrix $NT \times k$.
`W`	Spatial weight matrix ($N \times N$ or $NT \times NT$).
`N`	Integer: number of cross-sectional units.
`Time`	Integer: number of time periods.
`ndraw`	Integer: total MCMC draws (including burn-in).
`nomit`	Integer: burn-in draws to discard.
`prior`	List of prior hyperparameters (see `ols_panel` for common fields). Additional fields: **lflag** 0 = exact log-det, 1 = MC approximation (default). **order** Taylor order for MC log-det (default 50). **iter** MC iterations for log-det (default 30). **rmin, rmax** Bounds for ρ grid (default `c(-1,1)`). **lndet** Pre-computed log-det grid (2-column matrix) to reuse.

Details

$$y = X\beta + WX\theta + \text{FE} + \varepsilon$$

No spatial autoregressive parameter is estimated. Effects decomposition: direct = β, indirect = θ, total = $\beta + \theta$.

Value

An S3 object of class `"spmixW"` with OLS fields plus SLX effects (direct, indirect, total).

References

LeSage, J.P. and Pace, R.K. (2009). *Introduction to Spatial Econometrics*. Taylor & Francis/CRC Press.

Examples

```
set.seed(1)
N <- 30; Time <- 10
coords <- cbind(runif(N), runif(N))
W <- as.matrix(normw(make_knw(coords, k = 5, row_normalise = FALSE)))
Wbig <- kronecker(diag(Time), W)
X <- matrix(rnorm(N * Time * 2), ncol = 2)
WX <- Wbig %*% X
y <- X %*% c(1, -0.5) + WX %*% c(0.3, 0.2) + rnorm(N * Time)
res <- slx_panel(as.numeric(y), X, W, N, Time, ndraw = 3000, nomit = 1000)
print(res)
```

---

spmodel *Formula-Based Entry Point for Spatial Panel Models*

Description

A user-friendly interface that accepts a formula and data frame, automatically handles sorting, fixed-effects mapping, and dispatches to the appropriate low-level estimation function.

Usage

```
spmodel(
  formula,
  data,
  W,
  model = "sar",
  id,
  time = NULL,
```

```
  effects = "twoway",
  heteroscedastic = TRUE,
  rval = NULL,
  ndraw = 5500L,
  nomit = 1500L,
  thin = 1L,
  taylor_order = 6L,
  ...
)
```

Arguments

formula	A formula of the form y ~ x1 + x2 + ... An intercept is NOT included in the design matrix (fixed effects handle the level).
data	A data frame containing all variables referenced in the formula, plus the id and time columns.
W	A spatial weight matrix (N x N) or a list of M weight matrices for convex combination models. When a list is provided, the convex combination variant of the specified model is used automatically.
model	Character string specifying the model type. One of: "ols", "sar", "sdm", "sem", "sdem", "slx", "sar_bma", "sdm_bma", "sdem_bma".
id	Character: name of the column in data identifying cross-sectional units (regions).
time	Character or NULL: name of the column identifying time periods. If NULL, a cross-sectional model (T=1) is estimated with effects = "none".
effects	Character: fixed-effects specification. One of "none", "region", "time", "twoway". Default "twoway".
heteroscedastic	Logical: if TRUE (default), use heteroscedastic errors with rval = 5. If FALSE, homoscedastic.
rval	Numeric: degrees-of-freedom for heteroscedastic errors. Overrides heteroscedastic if provided. Default NULL.
ndraw	Integer: total MCMC draws. Default 5500.
nomit	Integer: burn-in draws. Default 1500.
thin	Integer: thinning interval. Default 1.
taylor_order	Integer: Taylor series order for convex combination models. Default 6.
...	Additional arguments passed to the prior list (e.g., beta_mean, beta_var, nu, d0).

Details

Auto-sorting: The data is sorted by time first, then by region within each time period, to match the package's required stacking order (all N regions in period 1, then all N in period 2, etc.). Users do not need to pre-sort their data.

W list detection: When `W` is a list of matrices, the convex combination variant of the model is used automatically (e.g., `"sar"` becomes `sar_conv_panel()`).

Cross-sectional mode: When `time = NULL`, the function automatically sets T=1 and `effects = "none"`.

Value

An S3 object of class `"spmixW"` or `"spmixW_bma"` with additional metadata:

formula The formula used.

response_name Name of the response variable.

predictor_names Names of predictor variables.

id_var, time_var Column names used for panel structure.

Examples

```
coords <- cbind(runif(80), runif(80))
W_geo <- make_knw(coords, k = 5)
W_trade <- make_knw(coords, k = 10)

panel <- simulate_panel(N = 80, T = 8,
                        W = list(W_geo, W_trade),
                        gamma = c(0.6, 0.4), rho = 0.5,
                        beta = c(1.5, -0.8), seed = 123)

res <- spmodel(y ~ x1 + x2, data = panel,
               W = list(geography = W_geo, trade = W_trade),
               model = "sar", id = "region", time = "year",
               effects = "twoway", ndraw = 8000, nomit = 2000)
print(res)
```

---

summary.spmixW *Summary Method for spmixW Objects*

Description

Summary Method for spmixW Objects

Usage

```
## S3 method for class 'spmixW'
summary(object, ...)
```

Arguments

`object`	An object of class `"spmixW"`.
`...`	Further arguments (ignored).

Value

Invisibly returns `object` with MCMC diagnostics appended.

---

summary.spmixW_bma *Summary Method for spmixW_bma Objects*

Description

Summary Method for spmixW_bma Objects

Usage

```
## S3 method for class 'spmixW_bma'
summary(object, ...)
```

Arguments

`object`	An object of class `"spmixW_bma"`.
`...`	Further arguments (ignored).

Value

Invisibly returns `object`.

---

tidy.spmixW *Tidy Method for spmixW Objects*

Description

Returns a data frame of estimated parameters with standard errors, test statistics, p-values, and credible intervals.

Usage

```
tidy.spmixW(x, effects = TRUE, ...)
```

Arguments

`x`	An object of class `"spmixW"`.
`effects`	Logical: if `TRUE` (default), include direct, indirect, and total effects rows in addition to raw coefficients.
`...`	Additional arguments (ignored).

Value

A data frame with columns: `term`, `estimate`, `std.error`, `statistic`, `p.value`, `conf.low`, `conf.high`, `type`.

`tidy.spmixW_bma` *Tidy Method for spmixW_bma Objects*

Description

Tidy Method for spmixW_bma Objects

Usage

```
tidy.spmixW_bma(x, ...)
```

Arguments

`x`	An object of class `"spmixW_bma"`.
`...`	Additional arguments (ignored).

Value

A data frame with BMA-averaged coefficients and model probabilities.

Index