Package: SpectralMM (via r-universe)

October 18, 2026

Type Package

Title Spectral Majorization for Statistical Estimation

Version 0.2.0

Description Fits generalized linear, asymmetric and robust regression models using spectral majorization and iterative quadratic subproblem solvers. A shared C++ computational core supports dense and sparse designs, automatic intercepts, optional Wald inference and detailed iteration diagnostics. Native code is compiled automatically during package installation.

License GPL-3

Encoding UTF-8

Depends R (>= 4.0.0)

Imports Rcpp, stats, utils

LinkingTo Rcpp, RcppEigen

Suggests Matrix

SystemRequirements C++17

NeedsCompilation yes

URL https://github.com/Xunjian-Li/SpectralMM

BugReports https://github.com/Xunjian-Li/SpectralMM/issues

Repository https://xunjian-li.r-universe.dev

Date/Publication 2026-10-18 02:18:04 UTC

RemoteUrl https://github.com/xunjian-li/spectralmm

RemoteRef HEAD

RemoteSha 01f3a5dc8577b472dcd786590e540f566313c033

RemoteSubdir R

Contents

SpectralMM-package 2
spectralmm_family 2
spectralmm_fit 3
spectralmm_methods 5

Index **8**

`SpectralMM-package` *Spectral Majorization for Statistical Estimation*

Description

A common C++ backend for generalized linear, asymmetric and robust regression, with iterative and Cholesky solvers, dense/sparse inputs, optional inference and iteration diagnostics.

Author(s)

Xun-Jian Li

See Also

`spectralmm_fit`

`spectralmm_family` *Non-GLM Loss Specifications*

Description

Lightweight frontend specifications for the existing non-GLM fitting implementation. Use these constructors with spectralmm_fit, not spectralmm_glm.

Usage

```
pseudo_huber(delta = 1)
expectile(q = 0.5)
smooth_quantile(q = 0.5, epsilon = 0.1)
student_t(nu = 4)
```

Arguments

`delta`	Positive finite pseudo-Huber scale.
`q`	Finite asymmetry level strictly between zero and one.
`epsilon`	Positive finite smoothing parameter.
`nu`	Positive finite Student-t degrees of freedom. The residual scale retains its existing default of one.

Details

Matrix and formula fits normalize these objects to the existing string and family_options representation. The public q maps to the internal tau; epsilon maps to smoothing. Supplying family_options together with an object is an error, even for NULL or an empty list. Existing string specifications remain supported unchanged, including custom Student-t sigma. All interfaces use delta=1 by default, including the legacy string path. To reproduce the earlier development default, explicitly set delta=1.345.

Value

A lightweight object of class spectralmm_family, containing a core family name and its options.

See Also

spectralmm_fit

Examples

```
set.seed(1)
dat <- data.frame(y=rnorm(100), x=rnorm(100))
m <- spectralmm_fit(y ~ x, data=dat, family=pseudo_huber(delta=1))
coef(m)
```

---

spectralmm_fit *Fit Statistical Models with SpectralMM*

Description

Matrix and formula interfaces sharing the same numerical core. Use the GLM interface for conventional family/link specifications and the generic interface for residual losses or legacy family identifiers.

Usage

```
spectralmm_glm(X, y = NULL, data = NULL, family = stats::gaussian(),
  link = NULL, start = NULL, rank = NULL, solver = "pcg",
  control = spectralmm_control(), ...)
spectralmm_fit(X, y = NULL, family = "gaussian", data = NULL,
  start = NULL, beta0 = NULL, intercept = NULL, rank = NULL,
  solver = "pcg", control = spectralmm_control(),
  weights = NULL, offset = NULL, ...)
spectralmm_control(...)
spectralmm_diagnostics(object)
```

Arguments

`X`	A numeric feature matrix, a sparse dgCMatrix, or a formula.
`y`	Numeric response vector for matrix input.
`data`	A data frame for formula input. Missing values are rejected.
`family`	For GLMs, an R family object (gaussian, binomial, poisson or Gamma), or a supported family name. Negative binomial with fixed theta uses the name `"negative_binomial"`. For generic fitting, a constructor such as `pseudo_huber(delta=1)`, or a legacy core family name or residual loss: smooth_quantile, expectile, pseudo_huber or student_t.
`link`	Link name for a named family; specify it within an R family object instead when using one. Supported pairs are Gaussian identity/log, binomial logit/probit, Poisson log, Gamma log/inverse and negative binomial log.
`start`, `beta0`	Initial coefficients including an intercept when present. beta0 is a compatibility alias; do not supply both.
`intercept`	NULL adds an intercept for matrix input. FALSE means a complete design matrix. Formula input controls its own intercept and rejects this argument.
`rank`	Retained spectral rank; NULL retains the existing numerical default.
`solver`	pcg, mm, cg, cho, cgls, crls, lsqr or lsmr. Legacy aliases spectral and cholesky remain accepted.
`control`	Advanced numerical settings from spectralmm_control. Omitted settings retain existing defaults. Duplicate direct and control options are rejected.
`weights`, `offset`	Not implemented by the numerical core. Non-NULL values are rejected, including formula offsets.
`object`	A fitted SpectralMM model.
`...`	Numerical settings (maxiter, inner_maxiter, krylovdim, ridge, floor, gtol, relgtol, eta_max, correction_tol, resid_tol, nesterov, preconditioner, krylov_rtol, krylov_atol, accept_negligible, accept_stalled, negligible_step_tol, step_reltol, stalled_relgtol); or fit arguments family_options, inference, inference_max_p, cov_type, level, use_t, dispersion, verbose, trace, penalize_intercept. Control accepts only numerical settings.

Details

The GLM binomial interface currently accepts binary zero/one responses, not proportions or two-column success/failure responses. Generic binomial fitting retains success counts with family_options=list(trials=...). Probit is a binomial link; the old probit core identifier remains a generic compatibility alias. Fixed negative binomial shape uses family_options=list(theta=...).

Formula fits preserve terms, factor levels, contrasts and coefficient names for new-data prediction. Both input routes call the same solver. Matrix input defaults to automatically adding one intercept. Starting vectors follow the resulting coefficient order.

Inference defaults to automatic only for at most 50 coefficients including the intercept. It also requires an unpenalized gradient-converged identifiable model with n greater than p. Setting inference=TRUE bypasses only the size cutoff. No covariance matrix is allocated merely for summary or diagnostics.

Deviance is unscaled. Gaussian log likelihood uses the maximum-likelihood scale RSS/n unless dispersion is supplied. Gamma log likelihood is evaluated at the supplied dispersion or Pearson dispersion with n-p denominator; this is not a profiled Gamma likelihood. The covariance dispersion policy is unchanged. Shape parameters remain fixed. Student-t offers a normalized location-scale log likelihood with fixed nu and sigma, but no deviance. Normalized likelihood is unavailable for the other residual losses and Tweedie. Verbose output displays LogLik where available, otherwise Objective (summed loss plus ridge). Gradient columns always refer to the optimization objective.

Value

A spectralmm object retaining the spectralmm_native compatibility class. It contains coefficients, family/link, formula metadata, cached linear predictors, fitted values, response residuals, inference and separate numerical diagnostics. The control constructor returns a validated options list; the diagnostics accessor returns a list.

See Also

spectralmm_methods

Examples

```
set.seed(1)
dat <- data.frame(x1=rnorm(100), x2=rnorm(100), y=rbinom(100,1,.5))
m <- spectralmm_glm(as.matrix(dat[c("x1","x2")]), dat$y,
                    family=binomial("logit"))
f <- spectralmm_glm(y ~ x1 + x2, data=dat, family=binomial("logit"))
coef(f)
summary(f)
head(predict(f, dat, type="response"))
spectralmm_diagnostics(f)$termination_reason
```

---

spectralmm_methods *Extract Results from a SpectralMM Fit*

Description

Coefficient, prediction, covariance, confidence interval and summary methods for fitted SpectralMM models.

Usage

```
## S3 method for class 'spectralmm'
predict(object, newdata = NULL, type = NULL,
  trials = NULL, offset = NULL, ...)
## S3 method for class 'spectralmm'
fitted(object, ...)
## S3 method for class 'spectralmm'
residuals(object, type = "response", ...)
```

```
## S3 method for class 'spectralmm'
nobs(object, ...)
## S3 method for class 'spectralmm'
deviance(object, ...)
## S3 method for class 'spectralmm'
logLik(object, ...)
## S3 method for class 'spectralmm'
summary(object, ...)
## S3 method for class 'spectralmm'
print(x, ...)
## S3 method for class 'summary.spectralmm'
print(x, digits = 6, max_rows = 20L, ...)

## S3 method for class 'spectralmm_native'
coef(object, ...)
## S3 method for class 'spectralmm_native'
predict(object, newdata, trials = NULL, ...)
## S3 method for class 'spectralmm_native'
vcov(object, ...)
## S3 method for class 'spectralmm_native'
confint(object, parm, level = .95, ...)
## S3 method for class 'spectralmm_native'
summary(object, ...)
## S3 method for class 'spectralmm_native'
print(x, ...)
## S3 method for class 'summary.spectralmm_native'
print(x, digits = 5, max_rows = 20L, ...)
```

Arguments

`object`	A fitted model.
`x`	A fitted model or summary, as appropriate.
`type`	Prediction: link or response; NULL defaults to link for spectralmm_glm and response for legacy spectralmm_fit. Residuals: only response is supported.
`newdata`	NULL for training predictions, a data frame for formula fits, or a numeric or sparse matrix with the same feature columns and ordering as training data, without an automatically fitted intercept column.
`trials`	Grouped binomial prediction trial counts, a positive scalar or vector. Defaults to the fitting value. Supply new counts for new rows when needed.
`offset`	Not implemented; a non-NULL value raises an error.
`parm`	Optional coefficient indices or names.
`level`	Confidence level between zero and one.
`digits`	Number of significant digits to print.
`max_rows`	Maximum coefficient rows to print. Full results remain in the summary.
`...`	Further arguments; the fitted-model print method forwards them to the summary print method.

Details

Coefficient tables use a t or z reference. Separate `wald_chisq` and `wald_p_value` components use an asymptotic chi-square reference; their p-values can differ from t-reference p-values.

Value

Coefficients (intercept first), response predictions, a covariance matrix, a confidence-interval matrix, or a summary list, depending on the method. Binomial predictions are expected counts. Non-GLM predictions are fitted locations: conditional expectiles for expectile loss, smoothed quantiles for smoothed quantile loss, and robust locations for Pseudo-Huber and Student-t. These are point predictions, not prediction intervals. Print methods return their input invisibly. Covariance and confidence interval methods signal an error when inference is unavailable.

See Also

`spectralmm_fit`

Index

coef.spectralmm_native
(spectralmm_methods), 5
confint.spectralmm_native
(spectralmm_methods), 5

deviance.spectralmm
(spectralmm_methods), 5

expectile (spectralmm_family), 2

fitted.spectralmm (spectralmm_methods), 5

logLik.spectralmm (spectralmm_methods), 5

nobs.spectralmm (spectralmm_methods), 5

predict.spectralmm
(spectralmm_methods), 5
predict.spectralmm_native
(spectralmm_methods), 5
print.spectralmm (spectralmm_methods), 5
print.spectralmm_native
(spectralmm_methods), 5
print.summary.spectralmm
(spectralmm_methods), 5
print.summary.spectralmm_native
(spectralmm_methods), 5
pseudo_huber (spectralmm_family), 2

residuals.spectralmm
(spectralmm_methods), 5

smooth_quantile (spectralmm_family), 2
SpectralMM (SpectralMM-package), 2
SpectralMM-package, 2
spectralmm_control (spectralmm_fit), 3
spectralmm_diagnostics
(spectralmm_fit), 3
spectralmm_family, 2
spectralmm_fit, *2*, 3, *3*, *7*
spectralmm_glm (spectralmm_fit), 3
spectralmm_methods, 5, *5*
student_t (spectralmm_family), 2
summary.spectralmm
(spectralmm_methods), 5
summary.spectralmm_native
(spectralmm_methods), 5

vcov.spectralmm_native
(spectralmm_methods), 5