Package: savvyGLM (via r-universe)

October 16, 2026

Type Package

Title Generalized Linear Models with Slab and Shrinkage Estimators

Version 0.1.4

Description Provides a flexible framework for fitting generalized linear models (GLMs) with slab and shrinkage estimators. Methods include the Stein estimator (St), Diagonal Shrinkage (DSh), Simple Slab Regression (SR), Generalized Slab Regression (GSR), Ledoit-Wolf Linear Shrinkage (LW), Quadratic-Inverse Shrinkage (QIS), and Shrinkage (Sh), all integrated into the iteratively reweighted least squares (IRLS) algorithm. This approach enhances estimation accuracy, convergence, and robustness in the presence of multicollinearity. The best-fitting model is selected based on the Akaike Information Criterion (AIC). Methods are related to methods described in Marschner (2011) <doi:10.32614/RJ-2011-012>, Asimit et al. (2025) <https://openaccess.city.ac.uk/id/eprint/35005/>, Ledoit and Wolf (2004) <doi:10.1016/S0047-259X(03)00096-4>, and Ledoit and Wolf (2022) <doi:10.3150/20-BEJ1315>.

License GPL (>= 3)

URL https://Ziwei-ChenChen.github.io/savvyGLM/

BugReports https://github.com/ziwei-chenchen/savvyGLM/issues

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.6.0)

Imports stats, MASS, glmnet, expm, parallel, glm2, utils, Matrix, CVXR

Suggests devtools, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs cmake libgmp3-dev make pkg-config libclang-dev

Repository https://ziwei-chenchen.r-universe.dev

Date/Publication 2026-05-10 12:27:31 UTC

RemoteUrl https://github.com/ziwei-chenchen/savvyglm

RemoteRef HEAD

RemoteSha 9e2bafbde66311b88e2d410dd95eab9f103e0201

Contents

savvy_glm.fit2 2
savvy_glm2 6

Index **10**

`savvy_glm.fit2` *Generalized Linear Models Fitting with Slab and Shrinkage Estimators*

Description

`savvy_glm.fit2` is a modified version of `glm.fit` that enhances the standard iteratively reweighted least squares (IRLS) algorithm by incorporating a set of shrinkage estimator functions. These functions (namely `St_ost`, `DSh_ost`, `SR_ost`, `GSR_ost`, `LW_est`, `QIS_est`, and optionally `Sh_ost`) provide alternative coefficient updates based on shrinkage principles. The function allows the user to select one or more of these methods via the `model_class` argument. Additionally, it optionally implements a robust optimization-based method to find starting values for fragile link functions.

Usage

```
savvy_glm.fit2(x, y, weights = rep(1, nobs),
                    model_class = c("St", "DSh", "SR", "GSR", "LW", "QIS", "Sh"),
                         start = NULL, etastart = NULL, mustart = NULL,
                         offset = rep(0, nobs), family = gaussian(),
                         control = list(), intercept = TRUE,
                         use_parallel = FALSE, use_robust_start = FALSE)
```

Arguments

`x`	A numeric matrix of predictors. As for `glm.fit`.
`y`	A numeric vector of responses. As for `glm.fit`.
`weights`	An optional vector of weights to be used in the fitting process. As for `glm.fit`.
`model_class`	A character vector specifying the shrinkage model(s) to be used. Allowed values are `"St"`, `"DSh"`, `"SR"`, `"GSR"`, `"LW"`, `"QIS"`, and `"Sh"`. If a single value is provided, only that method is run. If multiple values are provided, the function runs the specified methods in parallel (if `use_parallel = TRUE`) and returns the best one based on Akaike Information Criterion (AIC). When the user does not explicitly supply a value for `model_class`, the default is `c("St", "DSh", "SR", "GSR", "LW", "QIS")` (i.e. `"Sh"` is not considered).

start	Starting values for the parameters. As for glm.fit. If NULL and use_robust_start = TRUE, robust starting values may be calculated automatically (see Details).
etastart	Starting values for the linear predictor. As for glm.fit.
mustart	Starting values for the mean. As for glm.fit.
offset	An optional offset to be included in the model. As for glm.fit.
family	A description of the error distribution and link function to be used in the model. As for glm.fit.
control	A list of parameters for controlling the fitting process. As for glm.fit.
intercept	A logical value indicating whether an intercept should be included in the model, default is TRUE. As for glm.fit.
use_parallel	Logical. If TRUE, enables parallel execution of the fitting process. Defaults to TRUE. Set to FALSE for serial execution.
use_robust_start	Logical. If TRUE, uses an optimization-based approach (via the **CVXR** package) to calculate robust starting values for fragile link functions (e.g., "sqrt"). Defaults to FALSE to save computational time, as standard initialization works well for most typical datasets.

Details

savvy_glm.fit2 extends the classical Generalized Linear Model (GLM) fitting procedure by evaluating a collection of shrinkage-based updates during each iteration of the IRLS algorithm. When multiple shrinkage methods are specified in model_class (the default is c("St", "DSh", "SR", "GSR", "LW", "QIS"), thereby excluding "Sh" unless explicitly provided), if the user explicitly includes "Sh" (for example, model_class = c("St", "DSh", "SR", "GSR", "LW", "QIS", "Sh") or model_class = c("St", "Sh")), then the method Sh_ost is also evaluated.

The function can run these candidate methods in parallel (if use_parallel = TRUE). The final model is selected based on the lowest AIC. In cases where any candidate model returns an NA or non-finite AIC (which may occur when using quasi-likelihood families), the deviance is used uniformly as the model-selection criterion instead.

Robust Starting Values: In situations where the user does not provide start, etastart, or mustart, and the chosen family utilizes a fragile link function (specifically "log", "sqrt", "inverse", "1/mu^2", or bounded links like "logit"), standard GLM initialization can lead to infinite deviance or immediate divergence. If use_robust_start = TRUE, savvy_glm.fit2 employs a novel, data-driven optimization approach using the **CVXR** package to calculate stable starting coefficients (start). This process minimizes the squared difference between the linear predictor and a safely transformed target mean, enforcing positivity constraints strictly when required by the link function.

In essence, the function starts with initial estimates (either robustly generated or user-supplied) and iteratively updates the coefficients using the chosen shrinkage method(s). The incorporation of these shrinkage estimators aims to improve convergence properties and to yield more stable or parsimonious models under various data scenarios.

Shrinkage Estimators Used in the IRLS Algorithm:

Stein Estimator (St) Computes a single multiplicative shrinkage factor applied uniformly to all coefficients. The factor is chosen to minimize the overall mean squared error (MSE) of the OLS estimator. It is best suited when the predictors are similarly scaled so that uniform shrinkage is appropriate.

Diagonal Shrinkage (DSh) Extends the Stein estimator by calculating an individual shrinkage factor for each coefficient. Each factor is derived from the magnitude and variance of the corresponding coefficient, providing enhanced flexibility when predictor scales or contributions differ.

Simple Slab Regression (SR) Penalizes the projection of the OLS estimator along a fixed, predefined direction. It uses a closed-form solution under a slab (fixed-penalty) constraint, making it suitable when prior information suggests shrinking the coefficients along a known direction.

Generalized Slab Regression (GSR) Generalizes SR by incorporating multiple, data-adaptive shrinkage directions. It typically employs the leading eigenvectors of the design matrix as the directions, providing adaptive regularization that effectively handles multicollinearity.

Ledoit-Wolf Linear Shrinkage (LW) Implements linear shrinkage towards a one-parameter matrix where all variances are equal and all covariances are zero. This provides a well-conditioned estimator for large-dimensional covariance matrices, based on the methodology of Ledoit and Wolf (2004b).

Quadratic-Inverse Shrinkage (QIS) A nonlinear shrinkage estimator derived under Frobenius loss, Inverse Stein's loss, and Minimum Variance loss. It is highly effective for large covariance matrices and is based on the quadratic shrinkage approach of Ledoit and Wolf (2022).

Shrinkage Estimator (Sh) Computes a non-diagonal shrinkage matrix by solving a *Sylvester equation*. It transforms the OLS estimator to achieve a lower mean squared error. This estimator is evaluated only when `"Sh"` is explicitly included in the `model_class` argument.

Value

The value returned by `savvy_glm.fit2` has the same structure as the value returned by `glm.fit`. It includes the following components:

`coefficients`	the estimated coefficients.
`residuals`	the working residuals, that is the residuals in the final iteration of the IWLS fit.
`fitted.values`	the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.
`R`	the upper-triangular factor of the QR decomposition of the weighted model matrix.
`rank`	the numeric rank of the fitted linear model.
`qr`	the QR decomposition of the weighted model matrix.
`family`	the family object used.
`linear.predictors`	the final linear predictors.
`deviance`	the deviance of the final model.
`aic`	the AIC of the final model.
`null.deviance`	the deviance for the null model.
`iter`	the number of iterations used.
`weights`	the final weights used in the fitting process.
`prior.weights`	the weights initially supplied.

df.residual	the residual degrees of freedom.
df.null	the residual degrees of freedom for the null model.
y	the response vector used.
converged	a logical value indicating whether the IRLS iterations converged.
boundary	a logical value indicating whether the algorithm stopped at a boundary value.
time	the time taken for the fitting process.
chosen_fit	the name of the chosen fitting method based on AIC.

Author(s)

Ziwei Chen, Vali Asimit and Claudio Senatore
Maintainer: Ziwei Chen <Ziwei.Chen.3@citystgeorges.ac.uk>

References

Marschner, I. C. (2011). *glm2: Fitting Generalized Linear Models with Convergence Problems*. The R Journal, 3(2), 12–15. doi:10.32614/RJ2011012

Asimit, V., Avramescu, O., Chen, Z., Rivas, D., & Senatore, C. (2026). *GLM Solutions via Shrinkage*.

Asimit, V., Cidota, M. A., Chen, Z., & Asimit, J. (2025). *Slab and Shrinkage Linear Regression Estimation*. Retrieved from https://openaccess.city.ac.uk/id/eprint/35005/.

Ledoit, O. and Wolf, M. (2004). *A well-conditioned estimator for large-dimensional covariance matrices*. Journal of Multivariate Analysis, 88(2):365–411. doi:10.1016/S0047259X(03)000964

Ledoit, O. and Wolf, M. (2022). *Quadratic shrinkage for large covariance matrices*. Bernoulli, 28(3): 1519-1547. doi:10.3150/20BEJ1315

See Also

glm.fit, glm, glm.fit2

Examples

```
# Example 1: Standard Poisson regression with log link
set.seed(123)
n <- 100
p <- 5
X <- cbind(1, matrix(rnorm(n * p), n, p))
true_beta <- c(0.5, -0.2, 0.1, 0, 0, 0)

mu <- exp(X %*% true_beta)
y <- rpois(n, lambda = mu)

# Fit using savvy_glm.fit2 with the Stein (St) estimator
fit1 <- savvy_glm.fit2(x = X, y = y,
                       family = poisson(link = "log"),
                       model_class = "St")

print(fit1$coefficients)
```

```
# Example 2: Demonstrating Robust Starting Values for a Quadratic Link
eta <- abs(X %*% true_beta) + 1
mu_quad <- eta^2

# Generate Poisson data using the quadratic mean
y_quad <- rpois(n, lambda = mu_quad)
fit2 <- savvy_glm.fit2(x = X, y = y_quad,
                       family = poisson(link = "sqrt"),
                       model_class = c("SR", "St", "LW"),
                       use_parallel = TRUE,
                       use_robust_start = TRUE)

print(fit2$chosen_fit)
print(fit2$coefficients)
```

---

savvy_glm2 *Generalized Linear Models with Slab and Shrinkage Estimators*

Description

savvy_glm2 extends the classical glm2 function from the **glm2** package by embedding a set of shrinkage-based methods within the iteratively reweighted least squares (IRLS) algorithm. These shrinkage methods (implemented via savvy_glm.fit2) are designed to improve convergence and estimation accuracy. The user can specify one or more methods through the model_class argument. When multiple methods are provided (default is c("St", "DSh", "SR", "GSR", "LW", "QIS")), the function can evaluate them in parallel (controlled by the use_parallel argument) and selects the final model based on the lowest AIC.

Usage

```
savvy_glm2(formula, family = gaussian, data, weights,
            model_class = c("St", "DSh", "SR", "GSR", "LW", "QIS", "Sh"), subset,
                na.action, start = NULL, etastart, mustart, offset,
                control = list(...), model = TRUE,
                method = "savvy_glm.fit2", x = FALSE, y = TRUE,
           contrasts = NULL, use_parallel = FALSE, use_robust_start = FALSE, ...)
```

Arguments

formula	An object of class "formula": a symbolic description of the model to be fitted. As for glm2.
family	A description of the error distribution and link function to be used in the model. As for glm2.
data	An optional data frame, list or environment containing the variables in the model. As for glm2.
weights	An optional vector of weights to be used in the fitting process. As for glm2.

`model_class` A character vector specifying the shrinkage method(s) to be used in the underlying fitter `savvy_glm.fit2`. Allowed values are `"St"`, `"DSh"`, `"SR"`, `"GSR"`, `"LW"`, `"QIS"`, and `"Sh"`. If a single value is provided, only that method is executed. If multiple values are provided, the specified methods are evaluated (in parallel if `use_parallel = TRUE`) and the one with the lowest AIC is returned. By default, the value is `c("St", "DSh", "SR", "GSR", "LW", "QIS")`; note that the `"Sh"` method is considered only if explicitly included.

`subset` An optional vector specifying a subset of observations to be used in the fitting process. As for `glm2`.

`na.action` A function which indicates what should happen when the data contain NAs. As for `glm2`.

`start` Starting values for the parameters in the linear predictor. As for `glm2`. If `NULL` and `use_robust_start = TRUE`, robust optimization-based starting values may be calculated automatically for fragile link functions (see Details).

`etastart` Starting values for the linear predictor. As for `glm2`.

`mustart` Starting values for the vector of means. As for `glm2`.

`offset` An optional vector specifying a priori known component to be included in the linear predictor during fitting. As for `glm2`.

`control` A list of control parameters to pass to the iterative fitting process. As for `glm2`.

`model` A logical value indicating whether the model frame should be included as a component of the returned value. As for `glm2`.

`method` The method to be used in fitting the model. The default is `savvy_glm.fit2`. This method uses IRLS with custom optimization methods to ensure better convergence by evaluating different fitting methods and selecting the best one based on AIC. As in `glm2`, the alternative method "model.frame" returns the model frame and does no fitting.

`x` A logical value indicating whether the model matrix used in the fitting process should be returned as a component of the returned value. As for `glm2`.

`y` A logical value indicating whether the response vector used in the fitting process should be returned as a component of the returned value. As for `glm2`.

`contrasts` An optional list. See the contrasts.arg of `model.matrix.default`. As for `glm2`.

`use_parallel` A logical value specifying whether to evaluate multiple shrinkage methods in parallel. Defaults to `FALSE`. Setting this to `FALSE` forces sequential evaluation.

`use_robust_start`
Logical. If `TRUE`, uses an optimization-based approach (via the **CVXR** package) to calculate robust starting values for fragile link functions (e.g., "sqrt"). Defaults to `FALSE` to save computational time, as standard initialization works well for most typical datasets.

`...` Additional arguments to be passed to the low level regression fitting functions. As for `glm2`.

Details

`savvy_glm2` improves upon the standard Generalized Linear Model (GLM) fitting process by incorporating shrinkage estimator functions (`St_ost`, `DSh_ost`, `SR_ost`, `GSR_ost`, `LW_est`, `QIS_est`, and

optionally Sh_ost) within the IRLS algorithm. The function begins with initial parameter estimates and iteratively updates the coefficients using the specified shrinkage methods. When multiple methods are specified in model_class, they may be evaluated in parallel (if use_parallel = TRUE), and the final model is selected based on the lowest AIC. In cases where any candidate model returns an NA or non-finite AIC (such as when using quasi-likelihood families), the deviance is used uniformly as the selection criterion.

Robust Starting Values: In situations where the user does not provide start, etastart, or mustart, and the chosen family utilizes specific link functions (such as "log", "sqrt", "inverse", "1/mu^2", or "logit"), standard GLM initialization can sometimes lead to infinite deviance or divergence. If use_robust_start = TRUE, the underlying savvy_glm.fit2 automatically employs a data-driven optimization approach using the **CVXR** package to calculate stable starting coefficients.

Value

The value returned by savvy_glm2 has exactly the same structure as that returned by glm2, except for:

method	the name of the fitter function used, which by default is savvy_glm.fit2.
chosen_fit	the name of the chosen fitting method based on AIC.

Author(s)

Ziwei Chen, Vali Asimit and Claudio Senatore
Maintainer: Ziwei Chen <Ziwei.Chen.3@citystgeorges.ac.uk>

References

Marschner, I. C. (2011). *glm2: Fitting Generalized Linear Models with Convergence Problems*. The R Journal, 3(2), 12–15. doi:10.32614/RJ2011012

Asimit, V., Avramescu, O., Chen, Z., Rivas, D., & Senatore, C. (2026). *GLM Solutions via Shrinkage*.

Asimit, V., Cidota, M. A., Chen, Z., & Asimit, J. (2025). *Slab and Shrinkage Linear Regression Estimation*. Retrieved from https://openaccess.city.ac.uk/id/eprint/35005/.

Ledoit, O. and Wolf, M. (2004). *A well-conditioned estimator for large-dimensional covariance matrices*. Journal of Multivariate Analysis, 88(2):365–411. doi:10.1016/S0047259X(03)000964

Ledoit, O. and Wolf, M. (2022). *Quadratic shrinkage for large covariance matrices*. Bernoulli, 28(3): 1519-1547. doi:10.3150/20BEJ1315

See Also

glm, glm2, savvy_glm.fit2

Examples

```
set.seed(123)
n <- 200
p <- 5
X <- matrix(rnorm(n * p), n, p)
```

```
colnames(X) <- paste0("X", 1:p)

linear_predictor <- X %*% c(0.5, -0.5, 0.2, 0, 0)
prob <- 1 / (1 + exp(-linear_predictor))
y <- rbinom(n, 1, prob)

df <- data.frame(y = y, X)
fit1 <- savvy_glm2(y ~ ., data = df,
                   family = binomial(link = "logit"),
                   model_class = c("St", "DSh", "LW"),
                   use_parallel = FALSE)
print(fit1$chosen_fit)
print(coef(fit1))

eta <- abs(X %*% c(0.5, -0.2, 0.1, 0, 0)) + 1
mu_quad <- eta^2
y_quad <- rpois(n, lambda = mu_quad)
df_quad <- data.frame(y = y_quad, X)

fit2 <- savvy_glm2(y ~ ., data = df_quad,
                   family = poisson(link = "sqrt"),
                   model_class = c("SR", "St", "QIS"),
                   use_robust_start = TRUE,
                   use_parallel = FALSE)
print(fit2$chosen_fit)
print(coef(fit2))
```

Index

∗ **models**
savvy_glm.fit2, 2
∗ **regression**
savvy_glm.fit2, 2

glm, *5*, *8*
glm.fit, *2*, *3*, *5*
glm.fit2, *5*
glm2, *6–8*

savvy_glm.fit2, 2, *8*
savvy_glm2, 6