WLogit package

Wencan Zhu

Introduction

This package provides functions for implementing the variable selection approach in high-dimensional linear models called WLogit described in Zhu et al. (2022). This method is designed for taking into account the correlations that may exist between the predictors (columns of the design matrix). It consists in rewriting the initial high-dimensional logistic regression model to remove the correlation existing between the predictors and in applying the generalized Lasso criterion. We refer the reader to [1] for further details.

Given a design matrix $\mathbf{X}$ of size $n \times p$, $X_j^{(i)}$ corresponds to the measurement of the jth biomarker on sample i, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is the vector of effect size for each biomarker, with most components equal to zero. We assume that the binary response $y_1, y_2, ..., y_n$ are independent random variables having a Bernoulli distribution with parameter $\pi_{\boldsymbol{\beta}}(X^{(i)})$ ($y_i \sim Bernoulli(\pi_{\boldsymbol{\beta}}(X^{(i)}))$), where for all i in $\{1, \ldots, n\}$,

$$\pi_{\boldsymbol{\beta}}(X^{(i)}) = \frac{\exp\left(\sum_{j=1}^{p} \beta_j X_j^{(i)}\right)}{1 + \exp\left(\sum_{j=1}^{p} \beta_j X_j^{(i)}\right)}. \tag{1}$$

The rows of $\boldsymbol{X}$ are assumed to be the realizations of independent centered Gaussian random vectors having a covariance matrix equal to $\boldsymbol{\Sigma}$. The vector $\boldsymbol{\beta}$ is assumed to be sparse, *i.e.* a majority of its components is equal to zero. The goal of the WLoigt approach is to retrieve the indices of the nonzero components of $\boldsymbol{\beta}$, also called active variables.

Installation

To obtain WLogit, the simplest approach is to install it directly from the CRAN (Comprehensive R Archive Network) using the following command:

```
install.packages("WLogit", repos = "http://cran.us.r-project.org")
```

Alternatively, users can download the package source at http://cran.r-project.org/web/packages/WLogit/ and download the WLogit_2.0.tar.gz file.

Data generation

Correlation matrix $\boldsymbol{\Sigma}$

We consider a correlation matrix having the following block structure:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix} \tag{2}$$

where $\mathbf{\Sigma}_{11}$ is the correlation matrix of active variables with off-diagonal entries equal to α_1, $\mathbf{\Sigma}_{22}$ is the one of non active variables with off-diagonal entries equal to α_3 and $\mathbf{\Sigma}_{12}$ is the correlation matrix between active and non active variables with entries equal to α_2. In the following example: $(\alpha_1, \alpha_2, \alpha_3) = (0.3, 0.5, 0.7)$.

The first 10 variables are active variables among the $p = 500$ variables and $n = 100$.

```
p <- 500 # number of variables
d <- 10 # number of actives
n <- 100 # number of samples
actives <- c(1:d)
nonacts <- c(1:p)[-actives]
Sigma <- matrix(0, p, p)
Sigma[actives, actives] <- 0.3
Sigma[-actives, actives] <- 0.5
Sigma[actives, -actives] <- 0.5
Sigma[-actives, -actives] <- 0.7
diag(Sigma) <- rep(1,p)
```

Generation of $\boldsymbol{X}$ and $\boldsymbol{y}$

The design matrix is then generated with the correlation matrix $\mathbf{\Sigma}$ previously defined by using the function `mvrnorm` and the response variable $\boldsymbol{y}$ is generated according to model (1) where the non null components of $\boldsymbol{\beta}$ are equal to 1.

```
X <- MASS::mvrnorm(n = n, mu=rep(0,p), Sigma, tol = 1e-6, empirical = FALSE)
beta <- rep(0,p)
beta[actives] <- 1
pr <- CalculPx(X,beta=beta)
y <- rbinom(n,1,pr)
```

Variable selection with the package

With the previous $\boldsymbol{X}$ and $\boldsymbol{y}$, the function `WhiteningLogit` of the package can be used to select the active variables.

First, we load the WLogit package:

```
library(WLogit)
```

We fit the model using the most basic call to `WhiteningLogit`

```
mod <- WhiteningLogit(X = X, y = y)
```

"mod" is a list that contains all the relevant information of the fitted model for future use. Note that the argument `y` needs to be binary and only contains 0 or 1.

Additional arguments:

- `nlambda`: number of lambda to be considered, the default value is 50.
- `gamma`: parameter described in the paper. Its default value is 0.999.
- `maxit`: integer specifying the maximum number of steps for the iteration in the Iterative Re-weighted Least Square algorithm. Its default value is 100.

Outputs:

- `beta`: matrix of the estimations of β for all the λ considered.
- `beta.min`: estimation of β which maximizes the log-likelihood.
- `log.likelihood`: Log-likelihood for all the λ considered.
- `lambda`: All λ considered.

Estimation of β by $\widehat{\beta}(\lambda)$ which maximizes the log-likelihood

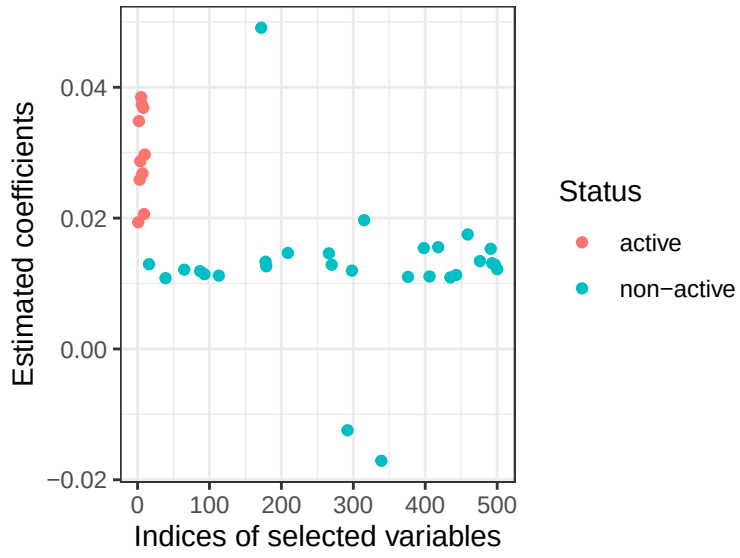
We show the first elements in estimated coefficients:

```
beta_min <- mod$beta.min
head(beta_min)
```

```
## [1] 0.01936466 0.03482722 0.02587475 0.02869973 0.03849732 0.03735274
```

Focusing on selected variables, we show which of them are truly active ones (red) and which are false positives (blue).

```
beta_min <- mod$beta.min
df_beta <- data.frame(beta_est=beta_min, Status = ifelse(beta==0, "non-active", "active"))
df_plot <- df_beta[which(beta_min!=0), ]
df_plot$index <- which(beta_min!=0)
ggplot2::ggplot(data=df_plot, mapping=aes(y=beta_est, x=index, color=Status))+geom_point()+
  theme_bw()+ylab("Estimated coefficients")+xlab("Indices of selected variables")
```



True Positive Rate: 1 (all active variables identified)

False Positive Rate: $28/490 = 0.0571429$

In this example, we have successfully selected all the true positives and only included 28 false positives out of the 490, which resulted in FPR equal to 0.057.

Compare to Lasso

Next we compare to Lasso by using the `glmnet` package with logistic regression. Glmnet is a package that fits a generalized linear model via penalized maximum likelihood. The regularization path is computed for the lasso or elastic-net penalty at a grid of values for the regularization parameter lambda. To select the optimized parameter, cross-validation is used and implemented by the function `cv.glmnet`. More details about this package can be found in its vignette (Friedman et al. (2010)).

```
library(glmnet)
cvfit = cv.glmnet(X, y, family = "binomial", type.measure = "class", intercept=FALSE)
```

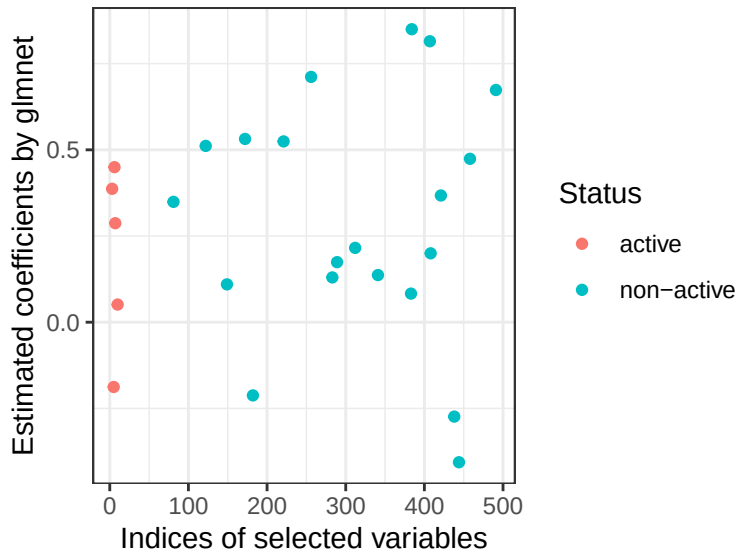
`lambda.min` is the value of lambda that gives minimum mean cross-validated error.

```
beta_lasso <- coef(cvfit, s = "lambda.min")
head(beta_lasso)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept)  .
## V1           .
## V2           .
## V3           0.3869045
## V4           .
## V5          -0.1879081
```

Finally we evaluate on variables selected by Lasso.

```
beta_lasso <- as.vector(beta_lasso)[-1]
df_beta <- data.frame(beta_est=beta_lasso, Status = ifelse(beta==0, "non-active", "active"))
df_plot <- df_beta[which(beta_lasso!=0), ]
df_plot$index <- which(beta_lasso!=0)
ggplot2::ggplot(data=df_plot, mapping=aes(y=beta_est, x=index, color=Status))+geom_point()+
  theme_bw()+ylab("Estimated coefficients by glmnet")+xlab("Indices of selected variables")
```



The selection accuracy of Lasso:

True Positive Rate: $5/10 = 0.5$

False Positive Rate: $20/490 = 0.0408163$

Lasso selected only five true positives including one with wrong sign.

We provide a compelling demonstration with this example, showcasing the effectiveness of our method (implemented in the WLogit package) in scenarios where the covariables exhibit high correlation and the irrepresentable condition is violated. In such challenging situations, our method outperforms the Lasso approach by successfully identifying all the true active cases with the correct sign. This outcome highlights the robustness and superiority of our method, even when faced with complex correlation patterns and violations of the irrepresentable condition.

References

[1] Zhu, W., Lévy-Leduc, C., & Ternès, N. (2022). Variable selection in high-dimensional logistic regression models using a whitening approach, Arxiv: 2206.14850.

[2] Friedman, J. H., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1–22.