EBglmnet Vignette

Anhui Huang and Dianting Liu

Nov. 30, 2015

Introduction

Installation

Quick Start

GLM Family

Prior, Hyperparameters and Epistasis

Introduction

Acronyms to be used EBglmnet is a package that implemented the empirical Bayesian Lasso (EBlasso) and Elastic Net (EBEN) method for generalized linear models (GLMs). Additionally, in `EBlasso`, two different prior distributions are also developed: one with two-level hierarchical Normal + Exponential prior (denoted as `NE`), and the other one with three-level Normal + Exponential + gamma prior (denoted as `NEG`). The following names should not be confused with the `lasso` and `elastic net` method in the comparison package `glmnet`:

`EBglmnet`: package that implements `EBlasso` and `EBEN` methods.

`EBlasso`: Empirical Bayesian method with `lasso` prior distribution, which includes two sets of prior distributions: `NE` and `NEG`.

`EBEN`: Empirical Bayesian method with `elastic net` prior distribution.

`lasso` prior: the hierarchical prior distribution that is equivalent with `lasso` penalty term when the marginal probability distribution for the regression coefficients is considered.

`elastic net` prior: the hierarchical prior distribution that is equivalent with `elastic net` penalty term when the marginal probability distribution for the regression coefficients is considered.

`EBlasso-NE`: `EBlasso` method having `NE` prior.

`EBlasso-NEG`: `EBlasso` method having `NEG` prior.

Generalized Linear Models (GLMs) In a GLM

$$\boldsymbol{\eta} = \mu\boldsymbol{I} + \mathbf{X}\boldsymbol{\beta},$$

where $\mathbf{X}$ is an $n \times p$ matrix containing p variables for n samples (p can be $\gg n$). $\boldsymbol{\eta}$ is an $n \times 1$ linear predictor and is related to the response variable $\boldsymbol{y}$ through a link function g: $\mathrm{E}(\mathbf{y}|\mathbf{X})=g^{-1}(\mu\boldsymbol{I} + \mathbf{X}\boldsymbol{\beta})$, and $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients. Depending on certain assumption of the data distribution on $\boldsymbol{y}$, the GLM is generally inferred through finding the set of model parameters that maximize the model likelihood function $p(\boldsymbol{y}|\mu, \boldsymbol{\beta}, \varphi)$, where φ denotes the other model parameter of the data distribution. However, such Maximum Likelihood (ML) approach is no longer applicable with $p \gg n$. With Bayesian Lasso and Bayesian elastic net (EN) prior distribution on $\boldsymbol{\beta}$, `EBglmnet` solves the problem by inferring a sparse posterior distribution

for $\hat{\boldsymbol{\beta}}$, which includes exactly zero regression coefficients for irrelevant variables and both posterior mean and variance for non-zero ones. Comparing to the `glmnet` package implementing Lasso and EN method, not only does `EBglmnet` provide features including both sparse outcome and hypothesis testing, simulation study and real data analysis in the reference papers also demonstrated the better performance in terms of Power of Detection, False Discovery Rate, as well as Power Detecting Group Effects when applicable. While mathematically details of the `EBlasso` and `EBEN` methods can be found in the reference papers, the principle of the methods and differences on the prior distributions will be briefly introduced here.

Lasso and its Bayesian Interpretation

Lasso applies a penalty term on the log likelihood function and solve for $\hat{\boldsymbol{\beta}}$ by maximizing the following penalized likelihood :

$$\hat{\boldsymbol{\beta}} = \arg_{\boldsymbol{\beta}} \max \left[\log p(\boldsymbol{y}|\mu, \boldsymbol{\beta}, \varphi) - \lambda||\boldsymbol{\beta}||_1\right],$$

The l_1 penalty term can be regarded as a mixture of hierarchical prior distribution:

$$\beta_j \sim N(0, \sigma_j^2), \sigma_j^2 \sim \exp(\lambda), j = 1, \ldots, p,$$

and maximizing the penalized likelihood function is equivalent to maximize the marginal posterior distribution of $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = \arg_{\boldsymbol{\beta}} \max \log p(\boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X}, \mu, \lambda, \varphi) \approx \arg_{\boldsymbol{\beta}} \max \log \int \left[p(\boldsymbol{y}|\mu, \boldsymbol{\beta}, \varphi) \cdot (2\pi)^{-p/2} |\mathbf{A}|^{1/2} \exp\{-\frac{1}{2}\boldsymbol{\beta}^T \mathbf{A} \boldsymbol{\beta}\} \cdot \prod_{j=1}^{p} \lambda \exp\{-\lambda \sigma_j^2\} \right] d\boldsymbol{\sigma}^2,$$

where $\mathbf{A}$ is a diagonal matrix with $\boldsymbol{\sigma}^{-2}$ on its diagonal. Of note, `lasso` integrates out the variance information $\boldsymbol{\sigma}^2$ and estimates a posterior mode $\hat{\boldsymbol{\beta}}$. The l_1 penalty ensures a sparse solution can be achieved.

Empirical Bayesian Lasso (`EBlasso`)

`EBglmnet` keeps the variance information integrated out in `lasso` while still enjoying the sparse property by taking a different and slightly complicated approach as showing below using `EBlasso-NE` as an example:

In contrary to the marginalization on $\boldsymbol{\beta}$, the first step in `EBlasso-NE` is to obtain a marginal posterior distribution for $\boldsymbol{\sigma}^2$:

$$p(\boldsymbol{\sigma}^2|\boldsymbol{y}, \mathbf{X}, \mu, \lambda, \varphi) = \int \left[p(\boldsymbol{y}|\mu, \boldsymbol{\beta}, \varphi) \cdot (2\pi)^{-p/2} |\mathbf{A}|^{1/2} \exp\{-\frac{1}{2}\boldsymbol{\beta}^T \mathbf{A} \boldsymbol{\beta}\} \cdot \prod_{j=1}^{p} \lambda \exp\{-\lambda \sigma_j^2\} + c \right] d\boldsymbol{\beta},$$

where c is a constant. While the integral in `lasso` is achieved through the conjugated normal + exponential (NE) prior, the integral in `EBlasso-NE` is completed through mixture of two normal distributions: $p(\boldsymbol{\beta}|\boldsymbol{\sigma}^2)$ and $p(\boldsymbol{y}|\mu, \boldsymbol{\beta}, \varphi)$, and the latter one typically is approximated to a normal distribution through Laplace approximation if itself is not a normal PDF. Then the estimate of $\hat{\boldsymbol{\sigma}}^2$ can be obtained by maximizing this marginal posterior distribution, which has the following form:

$$\hat{\boldsymbol{\sigma}}^2 = \arg_{\boldsymbol{\sigma}^2} \max \log p(\boldsymbol{\sigma}^2|\boldsymbol{y}, \mathbf{X}, \mu, \lambda, \varphi) = \arg_{\boldsymbol{\sigma}^2} \max \left[\log p(\boldsymbol{y}|\mu, \boldsymbol{\sigma}^2, \varphi, \lambda) - \sum_{j=1}^{p} \lambda \sigma_j^2 + c \right].$$

Given the constraint that $\boldsymbol{\sigma}^2 > 0$, the above equation is actually maximizing the l_1 penalized marginal likelihood function of $\boldsymbol{\sigma}^2$, which images the l_1 penalty in `lasso` with the beauty of producing a sparse solution for $\hat{\boldsymbol{\sigma}}^2$. Note that if $\hat{\sigma_j}^2 = 0$, $\hat{\beta}_j$ will also be zero and variable x_j will be excluded from the model. Finally, With the sparse estimate of $\hat{\boldsymbol{\sigma}}^2$, the posterior estimate of $\hat{\boldsymbol{\beta}}$ and other nuance parameters can then be obtained accordingly.

Hierarchical Prior Distributions in `EBglmnet`

Prior 1: EBlasso-NE

$$\beta_j \sim N(0, \sigma_j^2), \sigma_j^2 \sim \exp(\lambda), j = 1, \ldots, p$$

As illustrated above, assuming a Normal + Exponential hierarchical prior distribution on $\boldsymbol{\beta}$ (`EBlasso-NE`) will yield exactly the Lasso Prior. `EBlasso-NE` accommodates the properties of sparse solution and hypothesis testing given both the estimated mean and variance information in $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\sigma}}^2$. The NE prior is "peak zero and flat tails", which can select variables with relatively small effect size while shrinking most of non-effects to exactly zero. `EBlasso-NE` can be applied to natural population analysis when effect sizes are relatively small.

Prior 2: EBlasso-NEG In simulation and real data analysis, it is observed that the prior in `EBlasso-NE` has a relatively large probability mass on the nonzero tails, resulting in large number of non-zero small effects with large $p - values$. We further developed another well studied conjugated hierarchical prior distribution under the empirical Bayesian framework, the Normal + Exponential + Gamma (NEG) prior:

$$\beta_j \sim N(0, \sigma_j^2), \sigma_j^2 \sim \exp(\lambda), j = 1, \ldots, p\lambda \sim gamma(a, b)$$

Comparing to `EBlasso-NE`, the NEG prior has a larger probability centered at 0, and will only yield nonzero regression coefficients for effects having relatively large signal to noise ratio.

Prior 3: Elastic Net Prior for Grouping Effect Similar as `lasso`, `EBlasso` typically selects one variable out of a group of correlated variables. While `elastic net` was developed to encourage a grouping effect by incorporating an l_2 penalty term, `EBglmnet` implemented an innovative `elastic net` hierarchical prior:

$$\beta \sim N\left[0, (\lambda_1 + \tilde{\sigma_j}^{-2})^{-1}\right], \tilde{\sigma_j}^2 \sim generalized\ gamma(\lambda_1, \lambda_2), j = 1, \ldots, p.$$

The generalized gamma distribution has probability density function (PDF): $f(\tilde{\sigma_j}^2|\lambda_1, \lambda_2) = c(\lambda_1\tilde{\sigma_j}^2 + 1)^{-1/2}\exp\{-\lambda_2\tilde{\sigma_j}^2\}, j = 1, \ldots, p$, with c being a normalization constant. The property of this prior can be appreciated from the following aspects:

(1): $\lambda_1 = 0$ When $\lambda_1 = 0$ the generalized gamma distribution becomes an exponential distribution: $f(\tilde{\sigma_j}^2|\lambda_2) = c\exp\{-\lambda_2\tilde{\sigma_j}^2\}, j = 1, \ldots, p$, with $c = \lambda_2$, and the elastic net prior is reduced to the two level `EBlasso-NE` prior.

(2): $\lambda_1 > 0$ When $\lambda_1 > 0$ the generalized gamma distribution can be written as a shift gamma distribution having the following PDF: $f(\tilde{\sigma_j}^2|a, b, \gamma) = \frac{b^a}{\Gamma(a)}(\tilde{\sigma_j}^2 - \gamma)^{a-1}\exp\{-b(\tilde{\sigma_j}^2 - \gamma)\}$, where $a = 1/2, b = \lambda_2$, and $\gamma = -1/\lambda_1$. In (Huang A. 2015), it is proved that the marginal prior distribution for β_j can be obtained as $p(\beta_j) \propto \exp\{-\frac{\lambda_1}{2}\beta_j^2 - \sqrt{2\lambda_2}|\beta_j|\}$, which is equivalent with the `elastic net` method in `glmnet`.

(3): structure of $\boldsymbol{\sigma}^2$ and interpretation of the `elastic net` prior Note that the prior variance for the regression coefficients has this form: $\boldsymbol{\sigma}^2 = \tilde{\boldsymbol{\sigma}}^2/(\lambda_1\tilde{\boldsymbol{\sigma}}^2 + \boldsymbol{I})$. This structure seems counter intuitive at first glance. However, if we look at it from precision point of view, i.e., precision $\boldsymbol{\alpha} = \boldsymbol{\sigma}^{-2}$, and $\tilde{\boldsymbol{\alpha}} = \tilde{\boldsymbol{\sigma}}^{-2}$, then we have:

$$\boldsymbol{\alpha} = \lambda_1\boldsymbol{I} + \tilde{\boldsymbol{\alpha}}.$$

The above equation demonstrates that we actually decompose the precision of the normal prior into a fixed component λ_1 shared by all explanatory variables and a random component $\tilde{\boldsymbol{\alpha}}$ that is unique for each explanatory variable. This design represents the mathematical balance between the inter-group independence and intra-group correlation among explanatory variables, and is aligned with the objective of sparseness while encouraging grouping effects.

The empirical Bayesian elastic net (EBEN) in `EBglmnet` is solved similar as `EBlasso` using the aforementioned empirical Bayesian approach. Research studies presented in the reference papers demonstrated that `EBEN` has better performance comparing with `elastic net` in terms of Power of Detection, False Discovery Rate, and most importantly, Power of Detecting Groups.

`EBglmnet` Implementation and Usage

The `EBglmnet` algorithms use greedy coordinate descent, which successively optimizes the objective function over each parameter with others fixed, and cycles repeatedly until convergence. Key algorithms are implemented in C/C++ with matrix computation using the BLAS/LAPACK packages. Due to closed form solutions for $\hat{\boldsymbol{\sigma}}^2$ in all prior setups and other algorithmic and programming techniques, the algorithms can compute the solutions very fast.

We recommend to use `EBlasso-NEG` when there are a large number of candidate effects (eg. $\geq 10^6$ number of effects such as whole-genome epistasis analysis and GWAS), and use `EBEN` when there are groups of highly correlated variables.

The authors of `EBglmnet` are Anhui Huang and Dianting Liu. This vignette describes the principle and usage of `EBglmnet` in R. Users are referred to the papers in the reference section for details of the algorithms.

Installation

With Admin Permission on PC, `EBglmnet` can be installed directly from CRAN using the following command in R console:

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

which will download and install the package to the default directories. When Admin Permission is not immediately available, users can download the pre-compiled binary file at http://cran.r-project.org/web/packages/EBglmnet/index.html, and install it from local package.

Back to Top

Quick Start

We will give users a general idea of the package by using a simple example that demonstrates the basis package usage. Through running the main functions and examining the outputs, users may have a better idea on how the package works, what functions are available, which parameters to choose, as well as where to seek help. More details are given in later sections.

Let us first clear up the workspace and load the `EBglmnet` package:

```
rm(list = ls())
set.seed(1)
library(EBglmnet)
```

We will use an R built-in dataset `state.x77` as an example, which includes a matrix with 50 rows and 8 columns giving the following measurements in the respective columns: Population, Income, Illiteracy, Life Expectancy, Murder Rate, High School Graduate Rate, Days Below Freezing Temperature, and Land Area. The default model used in the package is the Gaussian linear model, and we will demonstrate it using Life Expectancy as the response variable and the remaining as explanatory variables. We create the input data as shown below, and users can load their own data and prepare variable `y` and `x` following this example.

```
varNames = colnames(state.x77);
varNames
```

```
## [1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
## [6] "HS Grad"    "Frost"      "Area"
```

```
y= state.x77[,"Life Exp"]
xNames = c("Population","Income","Illiteracy", "Murder","HS Grad","Frost","Area")
x = state.x77[,xNames]
```

We fit the model using the most basic call to `EBglmnet` with default prior

```
output = EBglmnet(x,y,hyperparameters = c(0.1, 0.1))
```

"output" is a list containing all the relevant information of the fitted model. Users can examine the output by directly looking at each elements in the list. Particularly, the sparse regression coefficients can be extracted as shown below:

```
glmfit = output$fit
variables = xNames[glmfit[,1,drop=FALSE]]
cbind(variables,as.data.frame(round(glmfit[,3:6,drop=FALSE],4)))
```

```
##   variables    beta posterior variance t-value p-value
## 1    Murder -0.2716              2e-04 19.1011       0
```

The hyperparameters in each of the prior distributions control the number of non-zero effects to be selected, and Cross-validation is perhaps the simplest and most widely used method in deciding their values. `cv.EBglmnet` is the main function to do cross-validation, which can be called using the following code.

```
cvfit = cv.EBglmnet(x, y)
```

```
## EBLASSO Linear Model, NEG prior,Epis:  FALSE ; 10 fold cross-validation
```

`cv.EBglmnet` returns a `cv.EBglmnet` object, which is a list with all the ingredients of the CV and the final fit results using CV selected optimal hyperparameters. We can view the CV results, selected hyperparameters and the corresponding coefficients. For example, CV using different hyperparameters and the corresponding prediction errors are shown below:

```
cvfit$CrossValidation
```

```
##           a    b Mean Square Error standard error
##  [1,]  0.01 0.01          3.466663      0.6217155
##  [2,]  0.05 0.05          3.483158      0.6260025
##  [3,]  0.10 0.10          3.592183      0.6462591
##  [4,]  0.50 0.50          3.716911      0.6422295
##  [5,]  1.00 1.00          3.737286      0.6403193
##  [6,] -0.01 0.01          3.487721      0.6408990
##  [7,] -0.10 0.01          3.547974      0.6427727
##  [8,] -0.20 0.01          3.486568      0.6400025
##  [9,] -0.30 0.01          3.433898      0.6263712
## [10,] -0.40 0.01          3.515311      0.6274390
## [11,] -0.50 0.01          3.464166      0.6238619
## [12,] -0.60 0.01          3.497567      0.6214415
## [13,] -0.70 0.01          3.400848      0.6386451
## [14,] -0.80 0.01          3.630952      0.5357177
## [15,] -0.90 0.01          4.373941      1.5046732
## [16,] -0.70 0.05          3.716178      0.5365876
## [17,] -0.70 0.10          3.783325      0.5157860
## [18,] -0.70 0.50          4.458618      1.4205773
```

The selected parameters and the corresponding fitting results:

```
cvfit$hyperparameters
```

```
##     a     b
## -0.70  0.01
```

```
cvfit$fit
```

```
##      locus1 locus2        beta posterior variance  t-value      p-value
## [1,]      4      4 -0.24209575       6.457118e-04 9.527255 9.796608e-13
## [2,]      5      5  0.03364319       1.515986e-05 8.640711 2.036060e-11
```

Back to Top

GLM Family

Two families of models have been developed in `EBglmnet`, the `gaussian` family and the `binomial` family, which are essentially different probability distribution assumptions on the response variable `y`.

Gaussian Model

`EBglmnet` assumes a Gaussian distribution on `y` by default, i.e., $p(\boldsymbol{y}|\mu,\boldsymbol{\beta},\varphi) = N(\mu\boldsymbol{I} + \mathbf{X}\boldsymbol{\beta}, \sigma_0^2\boldsymbol{I})$, where $\varphi = \sigma_0^2$ is the residual variance. In the above example, both $\hat{\mu}$ and $\hat{\sigma_0}^2$ are listed in the output:

```
output$Intercept
```

```
## [1] 72.88376
```

```
output$residual
```

```
## [1] 0.6912821
```

Binomial Model

If there are two possible outcomes in the response variable, a binomial distribution assumption on `y` is available in `EBglmnet`, which has $p(\boldsymbol{y}|\mu, \boldsymbol{\beta}, \varphi)$ following a binomial distribution and $\varphi \in \emptyset$. Same as the widely-used logistic regression model, the link function is $\eta_i = logit(p_i) = \log(\frac{Pr(y_i)=1}{1-Pr(y_i=1)}), i = 1, \ldots, n$. To run `EBglmnet` with binomial models, users need to specify the parameter `family` as `binomial`:

```
yy = y>mean(y);
output = EBglmnet(x,yy,family="binomial", hyperparameters = c(0.1, 0.1))
```

For illustration purpose, the above codes created a binary variable `yy` by set the cutoff at the mean Life Expectancy value.

Back to Top

Prior, Hyperparameters and Epistasis

The three sets of hierarchical prior distribution can be specified by `prior` option in `EBglmnet`. By default, `EBglmnet` assumes the `lassoNEG` prior, to change to other priors:

```
output = EBglmnet(x,yy,family="binomial", prior = "elastic net", hyperparameters = c(0.1, 0.1))
```

Note that the hyperparameters setup is associated with a specific prior. In `lasso` prior, only one hyperparameter λ is required, while in `elastic net` and `lassoNEG`, two hyperparameters need to be specified. For `EBEN` having the `elastic net` prior distribution, the two hyperparameters λ_1 and λ_2 are defined in terms of other two parameters $\alpha \in [0, 1]$ and $\lambda > 0$ same as in `glmnet` package, such that $\lambda_1 = (1 - \alpha)\lambda$ and $\lambda_2 = \alpha\lambda$. Therefore, users are asked to specify $hyperparameters = c(\alpha, \lambda)$.

In genetic and population analysis, sometimes it is interested in analyzing the interaction terms among the variables. `EBglmnet` provides a feature that can incorporate all pair-wise interactions into analysis, which is achieved by setting `Epis` as `TRUE`:

```
output = EBglmnet(x,yy,family="binomial", prior = "elastic net", hyperparameters = c(0.1, 0.1),Epis = TR
output$fit
```

```
##      locus1 locus2          beta posterior variance   t-value    p-value
## [1,]      4      4 -5.318341e-02       1.491454e-03 1.3771184 0.17473457
## [2,]      1      7  1.719555e-10       5.184252e-20 0.7552192 0.45373244
## [3,]      2      4 -8.894085e-06       6.408661e-11 1.1110091 0.27198645
## [4,]      3      4 -3.785224e-02       4.632023e-04 1.7587586 0.08486232
## [5,]      4      5 -3.100472e-04       2.447304e-07 0.6267348 0.53374233
## [6,]      4      6 -4.688616e-04       1.273471e-07 1.3138633 0.19501041
```

When `Epis = TRUE`, both p number of main effects and $p(p-1)/2$ number of interaction effects are considered in the model. In the output, `locus1` and `locus2` denote the pair of interaction variables, and if the numbers are the same, the corresponding effect is from a main effect. Users should be aware of the significant larger number variables considered (i.e., $p(p-1)/2$ more variables), and `EBglmnet` will need longer time when p is large for the program to finish the computation.

Back to Top

References

Anhui Huang, Shizhong Xu and Xiaodong Cai. (2015). Empirical Bayesian elastic net for multiple quantitative trait locus mapping Heredity, Vol. 114(1), 107-115.

Anhui Huang, Shizhong Xu and Xiaodong Cai. (2014a). Whole-genome quantitative trait locus mapping reveals major role of epistasis on yield of rice PLoS ONE, Vol. 9(1) e87330.

Anhui Huang, Eden Martin, Jeffery Vance, and Xiaodong Cai (2014b). Detecting genetic interactions in pathway-based genome-wide association studies Genetic Epidemiology, 38(4), 300-309.

Anhui Huang, Shizhong Xu and Xiaodong Cai. (2013). Empirical Bayesian LASSO-logistic regression for multiple binary trait locus mapping BMC Genetics, 14(1),5.

Xiaodong Cai, Anhui Huang and Shizhong Xu (2011). Fast empirical Bayesian LASSO for multiple quantitative trait locus mapping BMC Bioinformatics, 12(1),211.