HDBIG-SR Documentation

Release 1.0.0, 6/22/2014

Acknowledgements: NIH R01 LM011360 and NSF IIS-1117335.
Contact: Jingwen Yan (jingyan@umail.iu.edu) and/or Li Shen (shenli@iu.edu)
Question or bug reporting: The HDBIG team (hdbig@iu.edu)

1. Introduction

Recent advances in brain imaging and high throughput genotyping and sequencing techniques enable new approaches to study the influence of genetic variation on brain structure and function. HDBIG is a collection of software tools for high dimensional brain imaging genomics. These tools are designed to perform comprehensive joint analysis of heterogeneous imaging genomics data. HDBIG-SR is an HDBIG toolkit focusing on Sparse Regression. The current version includes matlab implementation of five sparse regression models. They can be applied to examine the association between genetic variations and imaging phenotypes. See below for a list of relevant papers.

- Yan J, Huang H, Kim S, Moore JH, Saykin AJ, Shen L, for the ADNI (2014) Joint identification of imaging and proteomics biomarkers of Alzheimer's disease using network-guided sparse learning. ***ISBI'14***: ***IEEE Int. Sym. on Biomedical Imaging***, pp 665-668, Beijing, China, 28 April - 2 May, 2014.
- Wang H, Nie F, Huang H, Kim S, Nho K, Risacher SL, Saykin AJ, Shen L, for the ADNI (2012) Identifying quantitative trait loci via group-sparse multi-task regression and feature selection: An imaging genetics study of the ADNI cohort. ***Bioinformatics***, 28(2):229-237. doi: 10.1093/bioinformatics/btr649
- Yan J, Risacher SL, Kim S, Simon JC, Li T, Wan J, Wang H, Huang H, Saykin AJ, Shen L, for the ADNI (2012) Multimodal neuroimaging predictors for cognitive performance using structured sparse learning. ***MBIA'12: MICCAI Workshop on Multimodal Brain Image Analysis***, Nice, France, October 1, 2012.

2. License

HDBIG-SR uses GNU General Public License (GPL). The license description is included in the software package. Please review and accept the license before installing HDBIG-SR via any source.

3. Download

Software

- Available at http://www.iu.edu/~hdbig/SR/

Documentation

- HTML: http://www.iu.edu/~hdbig/SR/HDBIG-SR-v1.0.0.html

- PDF: http://www.iu.edu/~hdbig/SR/HDBIG-SR-v1.0.0.pdf

4. Folder Structure and Demo Examples

The package "HDBIG-SR-v1.0.0.zip" consists of five subfolders.

- 00_data: Synthetic X, Y, and their prior structures (group and/or network)
- 01_example: Example functions for demonstration
- 02_data_preprocessing: Functions for data preprocessing
- 03_regression_code: Five regression functions (See "Methods" for details)
- 99_license: The license description.

All the functions described in the following "Methods" section are located in "03_regression_code". The current version only supports Matlab. For each of these functions, we have a corresponding example function for demonstration. These examples can be found under "01_example". Within each example, we perform the following steps

- Load synthetic data
- Data quality control (such as removing empty entries)
- Data Normalization (Let mean = 0 and standard deviation = 1)
- Running the corresponding regression model and return two outputs: trained weights and objective function values during iteration

5. Methods

In this package, five traditional and state-of-art regression models are included.

- ℓ_1 - Norm Regularization (Lasso)
- ℓ_1/ℓ_2 - Norm Regularization (Elastic Net)
- $\ell_{2,1}$ – Norm Regularization
- Group $\ell_{2,1}$ – Norm Regularization
- Network Guided $\ell_{2,1}$ – Norm Regularization

ℓ_1 - Norm Regularization (Lasso): Lasso is a traditional sparse regression model, which can help achieve sparse results by penalizing the ℓ_1–Norm.

Example usage:

- [W,obj] = R_Lasso(X, Y, para);

where X is n x c matrix and Y is n x d matrix. "para" controls the strength of the penalty term. Here n is subject number, c is predictor feature number and d is the response feature number. Trained weight and objective function values during iteration are returned in "W" and "obj" respectively.

ℓ_1/ℓ_2 - Norm Regularization (Elastic Net): Despite the overall sparsity, Lasso usually fails to handle correlated features. The ℓ_1–Norm mostly result in a random selection of correlated features with instable performance across trials. Elastic net bridges ℓ_1 and ℓ_2 norm together and aims to seek a balance in between, to achieve a less sparse but more stable pattern.

Example usage:

- [W,obj] = R_Elnet(X, Y, para);

where the parameters "X" and "Y" are the same as in Lasso. "para" is in range [0,1], which controls the strength percentage of ℓ_1 and ℓ_2 norm.

$\ell_{2,1}$ – Norm Regularization: While the high correlation within predictor features can be addressed by elastic net, interaction among response variables are usually ignored by performing each task separately. $\ell_{2,1}$ – Norm perfectly addresses this problem by coupling ℓ_1 and ℓ_2 norm in a different way, with ℓ_2 – norm among tasks and ℓ_1 norm still among features.

Example usage:

- [W,obj] = R_L21(X, Y, para);

where the parameters "X" and "Y" are the same as in Lasso. "para" controls the strength of the $\ell_{2,1}$ penalty term.

Group $\ell_{2,1}$ – Norm Regularization (GL21): As an extension of $\ell_{2,1}$ – norm, GL21 manages to incorporate the prior group structure of predictor variables, which achieves to yield both group-level and variable-level sparsity.

Example usage:

- [W,obj] = R_GL21(X, Y, group, para1,para2);

where the parameters "X" and "Y" are the same as in Lasso. "group" is a vector, indicating the group belongings of each predictor variable. "para1" controls the strength of group sparsity penalty, and "para2" controls the strength of $\ell_{2,1}$ penalty term.

Network Guided $\ell_{2,1}$ – Norm Regularization (NG-L21): Structured sparsity has received substantial attention in the past few years. While GL21 can only consider simple group structures, NG-L21 works as an advanced version of GL21, which takes more complicated network structure as input to guide the learning procedure.

Example usage:

- [W,obj] = R_NGL21(X, Y, network, para1,para2);

where the parameters "X" and "Y" are the same as in Lasso. "network" is a p x c matrix, indicating the network relationship among predictor variable. Each row in "network" indicates an edge, where only i^{th} and j^{th} element is not zero if it is connecting node I and j. "para1" controls the strength of network penalty, and "para2" controls the strength of $\ell_{2,1}$ penalty term.