Journal of Machine Learning Research 13 (2012) 1059-1062 Submitted 8/11; Revised 2/12; Published 4/12

The huge Package for High-dimensional Undirected Graph Estimation in R

Tuo Zhao TOURZHAO@JHU.EDU
Han Liu* HANLIU@CS.JHU.EDU
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA

Kathryn Roeder ROEDER@STAT.CMU.EDU
John Lafferty† LAFFERTY@CS.CMU.EDU
Larry Wasserman† LARRY@STAT.CMU.EDU
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA, 15213

Editor: Mikio Braun

Abstract

We describe an R package named huge which provides easy-to-use functions for estimating high dimensional undirected graphs from data. This package implements recent results in the literature, including Friedman et al. (2007), Liu et al. (2009, 2012) and Liu et al. (2010). Compared with the existing graph estimation package glasso, the huge package provides extra features: (1) instead of using Fortan, it is written in C, which makes the code more portable and easier to modify; (2) besides fitting Gaussian graphical models, it also provides functions for fitting high dimensional semiparametric Gaussian copula models; (3) more functions like data-dependent model selection, data generation and graph visualization; (4) a minor convergence problem of the graphical lasso algorithm is corrected; (5) the package allows the user to apply both lossless and lossy screening rules to scale up large-scale problems, making a tradeoff between computational and statistical efficiency.

Keywords: high-dimensional undirected graph estimation, glasso, huge, semiparametric graph estimation, data-dependent model selection, lossless screening, lossy screening

1. Overview

Undirected graphs is a natural approach to describe the conditional independence among many variables. Each node of the graph represents a single variable and no edge between two variables implies that they are conditional independent given all other variables. In the past decade, significant progress has been made on designing efficient algorithms to learn undirected graphs from high-dimensional observational data sets. Most of these methods are based on either the penalized maximum-likelihood estimation (Friedman et al., 2007) or penalized regression methods (Meinshausen and Bühlmann, 2006). Existing packages include glasso, Covpath and CLIME. In particu-

∗. Also in the Department of Biostatistics.
†. Also in the Department of Machine Learning.

lar, the `glasso` package has been widely adopted by statisticians and computer scientists due to its friendly user-inference and efficiency.

In this paper[1] we describe a newly developed `R` package named `huge` (High-dimensional Undirected Graph Estimation) coded in `C`. The package includes a wide range of functional modules and addresses some drawbacks of the graphical lasso algorithm. To gain more scalability, the package supports two modes of screening, lossless (Witten et al., 2011) and lossy screening. When using lossy screening, the user can select the desired screening level to scale up for high-dimensional problems, but this introduces some estimation bias.

2. Software Design and Implementation

The package `huge` aims to provide a general framework for high-dimensional undirected graph estimation. The package includes Six functional modules (M1-M6) facilitate a flexible pipeline for analysis (Figure 1).

M1. Data Generator: The function `huge.generator()` can simulate multivariate Gaussian data with different undirected graphs, including hub, cluster, band, scale-free, and Erdös-Rényi random graphs. The sparsity level of the obtained graph and signal-to-noise ratio can also be set up by users.

M2. Semiparametric Transformation: The function `huge.npn()` implements the nonparanormal method (Liu et al., 2009, 2012) for estimating a semiparametric Gaussian copula model.The nonparanormal family extends the Gaussian distribution by marginally transforming the variables. Computationally, the nonparanormal transformation only requires one pass through the data matrix.

M3. Graph Screening: The `scr` argument in the main function `huge()` controls the use of large-scale correlation screening before graph estimation. The function supports the lossless screening (Witten et al., 2011) and the lossy screening. Such screening procedures can greatly reduce the computational cost and achieve equal or even better estimation by reducing the variance at the expense of increased bias.

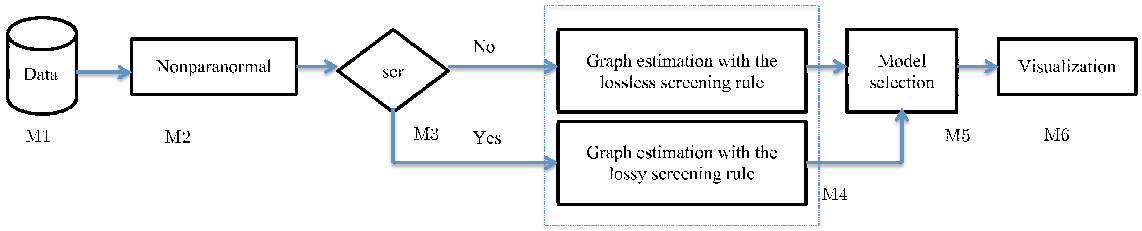

Figure 1: The graph estimation pipeline.

M4. Graph Estimation: Similar to the `glasso` package, the `method` argument in the `huge()` function supports two estimation methods: (i) the neighborhood pursuit algorithm (Meinshausen and Bühlmann, 2006) and (ii) the graphical lasso algorithm (Friedman et al., 2007). We apply the coordinate descent with active set and covariance update, as well as other tricks suggested in Friedman et al. (2010). We modified the warm start trick to address the potential divergence problem of the graphical lasso algorithm (Mazumder and Hastie, 2011). The code is also memory-optimized using the sparse matrix data structure when estimating and storing full regularization paths for large

1. This paper is only a summary of the package `huge`. For more details please refer to the online vignette.

data sets. we also provide a complementary graph estimation method based on thresholding the sample correlation matrix, which is computationally efficient and widely applied in biomedical research.

M5. Model Selection: The function `huge.select()` provides two regularization parameter selection methods: the stability approach for regularization selection (StARS) (Liu et al., 2010); and rotation information criterion (RIC). We also provide a likelihood-based extended Bayesian information criterion.

M6. Graph Visualization: The plotting functions `huge.plot()` and `plot()` provide visualizations of the simulated data sets, estimated graphs and paths. The implementation is based on the `igraph` package.

3. User Interface by Example

We illustrate the user interface by analyzing a stock market data which we contribute to the `huge` package. We acquired closing prices from all stocks in the S&P 500 for all the days that the market was open between Jan 1, 2003 and Jan 1, 2008. This gave us 1258 samples for the 452 stocks that remained in the S&P 500 during the entire time period.

```
> library(huge)
> data(stockdata)                                          # Load the data
> x = log(stockdata$data[2:1258,]/stockdata$data[1:1257,])  # Preprocessing
> x.npn = huge.npn(x, npn.func="truncation")               # Nonparanormal
> out.npn = huge(x.npn,method = "glasso", nlambda=40,lambda.min.ratio = 0.4)
```

Here the data have been transformed by calculating the log-ratio of the price at time t to price at time $t-1$. The nonparanormal transformation is applied to the data, and a graph is estimated using the graphical lasso (the default is the Meinshausen-Bühlmann estimator). The program automatically sets up a sequence of 40 regularization parameters and estimates the graph path. The lossless screening method is applied by default.

4. Performance Benchmark

To compare `huge` with `glasso` (ver 1.4), we consider four scenarios with varying sample sizes n and dimensionality d, as shown in Table 1. We simulate the data from a normal distribution with the Erdös-Rényi random graph structure (sparsity 1%). Timings (in seconds) are computed over 10 values of the corresponding regularization parameter, and the range of regularization parameters is chosen so that each method produced approximately the same number of non-zero estimates. The convergence threshold of both `glasso` and `huge` is chosen to be 10^{-4}. For these simulations, `CLIME` (ver 1.0) and `Covpath` (ver 0.2) were unable to obtain timing results due to their numerical instability.

For the neighborhood pursuit, we can see that `huge` achieves the best performance. In particular, when the lossy screening rule is applied, `huge` automatically reduces each individual lasso problem from the original dimension d to the sample size n, therefore a better efficiency can be achieved when d is much larger than n. Based on our experiments, the speed up due to the lossy screening rule can be up to more than 500%.

Method	$d = 1000$ $n = 100$	$d = 2000$ $n = 150$	$d = 3000$ $n = 200$	$d = 4000$ $n = 300$
huge-neighborhood pursuit (lossy)	3.246 (0.147)	13.47 (0.665)	35.87 (0.97)	247.2 (14.26)
huge-neighborhood pursuit	4.240 (0.288)	42.41 (2.338)	147.9 (4.102)	357.8 (28.00)
glasso-neighborhood pursuit	37.23 (0.516)	296.9 (4.533)	850.7 (8.180)	3095 (150.5)
huge-graphical lasso (lossy)	39.61 (2.391)	289.9 (17.54)	905.6 (25.84)	2370 (168.9)
huge-graphical lasso (lossless)	47.86 (3.583)	328.2 (30.09)	1276 (43.61)	2758 (326.2)
glasso-graphical lasso	131.9 (5.816)	1054 (47.52)	3463 (107.6)	8041 (316.9)

Table 1: Experimental Results

Unlike the neighborhood pursuit, the graphical lasso estimates the inverse covariance matrix. The screening rule (Witten et al., 2011) greatly reduces the computation required by the graphical lasso algorithm and gains an extra performance boost.

5. Summary and Acknowledgement

We developed a new package named huge for high dimensional undirected graph estimation. The package is complementary to the existing glasso package by providing extra features and functional modules. We plan to maintain and support this package in the future. Tuo Zhao is partially supported by the Google Summer of Code program 2011. Han Liu, John Lafferty, and Larry Wasserman are supported by NSF grant IIS-1116730 and AFOSR contract FA9550-09-1-0373. Kathryn Roeder is supported by National Institute of Mental Health grant MH057881.

References

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2010.

H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10:2295–2328, 2009.

H. Liu, K. Roeder, and L. Wasserman. Stability approach to regularization selection for high dimensional graphical models. *Advances in Neural Information Processing Systems*, 2010.

H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. High dimensional semiparametric gaussian copula graphical models. Technical report, Johns Hopkins University, 2012.

R. Mazumder and T. Hastie. The graphical lasso: New insights and alternatives. Technical report, Stanford University, 2011.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

D. Witten, J. Friedman, and Noah Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, to appear, 2011.