LASSO-penalized mixture of linear regressions model (LMLR) user manual

Luke R. Lloyd-Jones

Last updated March 25, 2016

Contents

1 Overview

Variable selection is an old and pervasive problem in regression analysis and has been widely discussed because of this; see George (2000) and Greene (2003, Ch. 8) for classical introductions to the topic, and see Hastie et al. (2009, Ch. 3) and Izenman (2008, Ch. 5) for some modern perspectives. In recent years, regularization has become popular in the statistics and machine learning literature, stemming from the seminal paper of Tibshirani (1996) on the least absolute shrinkage and selection operator (LASSO). A recent account of the literature regarding the LASSO and related regularization methods can be found in Buhlmann and van de Geer (2011). The mixture of linear regressions (MLR) for modeling heterogeneous data was first considered in Quandt (1972). The introduction of the EM (expectation–maximization) algorithm by Dempster et al. (1977) made such models simpler to estimate in a practical setting. Subsequently, MLR models became more popular; see DeSarbo and Cron (1988), De Veaux (1989), and Jones and McLachlan (1992) for example.

The LASSO-penalized MLR model (L-MLR) was considered in Khalili and Chen (2007) among a class of other regularization methods for the selection problem in the fixed number of variables setting. The L-MLR was then generalized to the divergent number of variables setting in Khalili and Lin (2013), and to the mixture of experts setting in Khalili (2010). Furthermore, Stadler et al. (2010) (see also Buhlmann and van de Geer (2011, Sec. 9.2)) considered an alternative parameterization of the L-MLR to Khalili and Chen (2007), and suggested a modified regularization expression. An alternative modified grouped LASSO criterion (Yuan and Lin, 2006) was suggested for regularization of the MLR problem in Hui et al. (2015). A recent review of the literature regarding the variable selection problem in MLR models can be found in Khalili (2011).

This program implements a new algorithm for the maximum penalized-likelihood (MPL) estimation of L-MLR models. This algorithm is constructed via the MM (minorization–maximization) algorithm paradigm of Lange (2013, Ch. 8). Such a construction allows for some desirable features such as coordinate-wise updates of parameters, monotonicity of the penalized likelihood sequence, and global convergence of the estimates to a stationary point of the penalized log-likelihood function. These three features are missing in the approximate-EM algorithm presented in Khalili and Chen (2007). Previously, MM algorithms have been suggested for the regularization of regression models in Hunter and Li (2005), where they are noted to be numerically stable. Coordinate-wise updates of parameters in LASSO-type problems was considered in Wu and Lange (2008), who also noted such updates to be fast and stable when compared to alternative algorithms. Furthermore, Stadler et al. (2010) also consider a coordinate-wise update scheme in their generalized EM algorithm, although the global convergence properties of the algorithm could only be established for the MPL estimation of a modified case of the L-MLR model with a simplified penalization function.

The difficulty in producing a globally convergent algorithm for the MPL estimation of the L-MLR model, which led both Khalili and Chen (2007) and Stadler et al. (2010) to utilize approximation schemes, is due to the intractability of the problem of updating the mixture model mixing proportions in the maximization-step of their respective algorithms. In our algorithm, we solve this issue by showing that it can be converted into a polynomial root finding problem. Our solution to this problem involves a polynomial basis conversion that is interesting in its own right. Aside from the new algorithm, we also consider the use of the L-MLR as a screening mechanism in a two-step procedure, as suggested in Buhlmann and van de Geer (2011, Sec. 2.5). Here, the L-MLR model is used to select the variable subset to include in an MLR model that is estimated in the second stage. This procedure allows for the adaptation of available asymptotic results for MLR models, such as those of Nguyen and McLachlan (2015), in order to obtain consistency and asymptotically normal parameter estimators. Optimization of the LASSO tuning parameter vector $\boldsymbol{\lambda}$ via derivative free numerical methods is also explored as an alternative to exhaustive grid search.

2 Mixture of Linear Regressions Model

Let $Y_1, ..., Y_n \in \mathbb{R}$ be an independent and identically distributed (IID) random sample that is dependent on corresponding covariate vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \in \mathbb{R}^p$, and let Z_t be a latent categorical random variable $(t = 1, ..., n)$ such that $z_t \in \{1, ..., g\}$, where $\mathbb{P}(Z_t = i) = \pi_i > 0$ and $\sum_{i=1}^{g} \pi_i = 1$. The MLR model can be defined via the conditional probability density characterization

$$f\left(y_t \mid \boldsymbol{x}_t, Z_t = i; \boldsymbol{\theta}\right) = \phi\left(y_t; \alpha_i + \boldsymbol{x}_t^T \boldsymbol{\beta}_i, \sigma_i^2\right),$$

which implies the marginal probability density characterization

$$f\left(y_i \mid \boldsymbol{x}_t; \boldsymbol{\theta}\right) = \sum_{i=1}^{g} \pi_i \phi\left(y_t; \alpha_i + \boldsymbol{x}_t^T \boldsymbol{\beta}_i, \sigma_i^2\right). \tag{1}$$

Here, $\phi\left(y; \mu, \sigma^2\right)$ is a normal density function with mean μ and variance σ^2, and we say that $\phi\left(y_t; \alpha_i + \boldsymbol{x}_t^T \boldsymbol{\beta}_i, \sigma_i^2\right)$ is the ith mixture component density. The vectors $\boldsymbol{\beta}_i = \left(\beta_{i1}, ..., \beta_{ip}\right)^T \in \mathbb{R}^p$, and scalars $\alpha_i \in \mathbb{R}$ and $\sigma_i^2 > 0$ are the specific regression coefficients, intercepts, and variances of the ith component density, respectively. We put all of the parameter components into the parameter vector $\boldsymbol{\theta} = \left(\boldsymbol{\pi}^T, \boldsymbol{\alpha}^T, \boldsymbol{\beta}^T, \boldsymbol{\sigma}^T\right)^T$, where $\boldsymbol{\pi} = \left(\pi_1, ..., \pi_g\right)^T$, $\boldsymbol{\alpha} = \left(\alpha_1, ..., \alpha_g\right)^T$, $\boldsymbol{\beta} = \left(\boldsymbol{\beta}_1^T, ..., \boldsymbol{\beta}_g^T\right)^T$, and $\boldsymbol{\sigma} = \left(\sigma_1^2, ..., \sigma_g^2\right)^T$.

Upon observation of a sample $y_1, ..., y_n$ with covariates $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ arising from an MLR with unknown parameter $\boldsymbol{\theta}_0 = \left(\boldsymbol{\pi}_0^T, \boldsymbol{\alpha}_0^T, \boldsymbol{\beta}_0^T, \boldsymbol{\sigma}_0^T\right)^T$, if no additional assumptions are made regarding the nature of the regression coefficients $\boldsymbol{\beta}_0$, the parameter vector can be estimated by the ML estimator $\tilde{\boldsymbol{\theta}}_n$, where $\tilde{\boldsymbol{\theta}}_n$ is an appropriate local maximizer of the log-likelihood function for the MLR model

$$\mathcal{L}_n\left(\boldsymbol{\theta}\right) = \sum_{t=1}^{n} \log \sum_{i=1}^{g} \pi_i \phi\left(y_t; \alpha_i + \boldsymbol{x}_t^T \boldsymbol{\beta}_i, \sigma_i^2\right).$$

2.1 LASSO-penalized MLR

Suppose that it is known that $\boldsymbol{\beta}_0$ is sparse, in the sense that some or many elements of $\boldsymbol{\beta}_0$ are exactly equal to zero. The estimates for the zero elements of $\boldsymbol{\beta}_0$, obtained via $\tilde{\boldsymbol{\theta}}_n$, will tend to be close to zero but will not shrink exactly to zero, and thus cannot be completely excluded from the model without the use of some other elimination techniques, such as via hypothesis testing. One method for simultaneously shrinking insignificant regression coefficients to zero and estimating the parameter vector $\boldsymbol{\theta}_0$, as suggested by Khalili and Chen (2007), is to estimate the L-MLR by computing the MPL estimator $\hat{\boldsymbol{\theta}}_n$, where $\hat{\boldsymbol{\theta}}_n$ is an appropriate local maximizer of the LASSO-penalized log-likelihood function for the MLR model

$$\mathcal{F}_n\left(\boldsymbol{\theta}\right) = \mathcal{L}_n\left(\boldsymbol{\theta}\right) - \mathcal{P}_n\left(\boldsymbol{\theta}\right). \tag{2}$$

Here,

$$\mathcal{P}_n(\boldsymbol{\theta}) = \sum_{i=1}^{g} \pi_i \sum_{j=1}^{p} \lambda_{in} \mid \beta_{ij} \mid \tag{3}$$

is the mixture LASSO penalty function, where $\lambda_{in} = n^{1/2}\gamma_{in}$ and $\gamma_{in} \geq 0$ are sequences of penalizing constants that are can be set to obtain a desired level of sparsity in the model. We note that $\tilde{\boldsymbol{\theta}}_n$ and $\hat{\boldsymbol{\theta}}_n$ are equivalent if $\lambda_{in} = 0$ for each i.

We now proceed to construct an MM algorithm for the MPL estimation of the L-MLR model. In order to produce an algorithm that is globally convergent, we follow the tactic of Hunter and Li (2005) and consider instead an appropriate local maximizer to the ϵ-approximate LASSO-penalized log-likelihood function

$$\mathcal{F}_{n,\epsilon}(\boldsymbol{\theta}) = \mathcal{L}_n(\boldsymbol{\theta}) - \mathcal{P}_{n,\epsilon}(\boldsymbol{\theta}), \tag{4}$$

where

$$\mathcal{P}_{n,\epsilon}(\boldsymbol{\theta}) = \sum_{i=1}^{g} \pi_i \sum_{j=1}^{p} \lambda_{in} \sqrt{\beta_{ij}^2 + \epsilon^2} \tag{5}$$

for some small $\epsilon > 0$. Similarly to Hunter and Li (2005, Prop. 3.2), we can show that $\mid \mathcal{F}_{n,\epsilon}(\boldsymbol{\theta}) - \mathcal{F}_n(\boldsymbol{\theta}) \mid \rightarrow 0$ uniformly as $\epsilon \rightarrow 0$, over any compact subset of the parameter space. The analysis of (4) instead of (2) is advantageous since its differentiability allows for the simple application of a useful global convergence theorem.

3 Download and installation

You can download the latest version of the BOLT-LMM software at:

`https://github.com/lukelloydjones/LMLR`

3.1 Change log

- Initial release April, 2016

The `LMLR` download package contains a standalone (i.e., statically linked) 64-bit OSX executable, `lmlr`, which we have tested on several OSX systems. This static executable requires no further installation. If you wish to compile your own version of the LMLR software from the source code (in the `src/`

subdirectory), you will need to install the library dependencies (which can be easily done with `homebrew` on OSX) and you will need to make appropriate path modifications to the `Makefile`:

- Library dependencies:
  - Armadillo C++ linear algebra library numerical libraries `http://arma.sourceforge.net/`.
  - Boost C++ libraries
- Makefile:
  - Paths to libraries need to be modified appropriately.
  - All versions of the current program were compiled with gcc version 4.9 from the GNU compiler collection.

3.2 Running LMLR

To run the `lmlr` executable, you can invoke it via `./lmlr` on the Linux or UNIX command line (within the install directory) with parameters following with spaces separating them. The best way to run the program is with the parameter file `lmlr_submit.sh`. The contents of this file and its use are outlined below.

3.3 Examples

The `examples/` subdirectory contains the data and bash scripts to run two examples, which demonstrate the basic use of `lmlr`.

The first example in `simulation/` contains the bash script `run_example_1.sh`, which will execute the program on a simulated data set of size $(n, p) = (200, 100)$. This example can be invoked, after necessary adjustment of folder paths, by typing `./run_example_2.sh` at the UNIX or Linux command line.

The second example is contained in the folder `baseball/` and has the bash script `run_example_2.sh`. The execution of this bash script demonstrates `lmlr` on the baseball salary data from Journal of Statistics Education (`www.amstat.org/publications/jse`), and was used in Lloyd-Jones et al. (2016); Khalili and Chen (2007). The folder also contains a description of the data (`baseball.txt`), which outlines the covariates used and their meaning, the original data in CSV format (`baseball_dat.csv`), and an R script for processing the data and results generated (`baseball_process.R`).

4 Computing requirements

4.1 Operating system

At the current time we have compiled and tested LMLR on Linux and Unix computing environments; however, the source code is available if you wish to try compiling LMLR for a different operating system.

4.2 Memory

Memory profiling will be updated in the next version of the user manual. At this point the program has not exceeded the memory capacity of 16GB for problems as large as $(n, p) = (1000, 10000)$.

4.3 Running time

A full running time profile will be updated in the next version of the user manual. At this point the program requires approximately 11 hrs for a $(n, p) = (500, 1000)$ problem.

5 Input/output file conventions

The `lmlr` program requires file input to be in comma separated values (CSV) format with no headers or row names. The $\boldsymbol{X}$ data matrix should be of size $n \times p$. The $\boldsymbol{Y}$ data vector should be of size $n \times 1$. The program also requires the specification of a initial $\boldsymbol{\beta}_0$ vector of size $p \times g$, which should reside in the path to the output directory. Files are written out in plain text CSV format.

6 Input

6.1 Predictor matrix

The predictor $\boldsymbol{X}$ data matrix contains the information on the set of predictors for the analysis. It should be of size $n \times p$ and at this stage cannot contain any missing data. It is also assumed that the dimensions and rows in the predictor and phenotype align. It is also desirable to column standardise the $\boldsymbol{X}$ matrix so that the columns have mean 0 and variance 1. It is not required to add a set of 1's to the $\boldsymbol{X}$ to indicate an intercept term. At this stage is should be in CSV format with no row or column identifiers.

6.2 Phenotype vector

The phenotype vector $\boldsymbol{Y}$ data matrix contain the information on the dependent variable or phenotype for the analysis. It should be of size $n \times 1$ and at this stage cannot contain any missing data. It is also assumed that the dimensions and rows in the predictor and phenotype align. At this stage is should be in CSV format with no row or column identifiers.

6.3 Initial regression parameter matrix

For many problems the $\boldsymbol{\beta}$ matrix is very large and thus require the use of a starting set of parameter estimates for the component regression coefficients denoted $\boldsymbol{\beta}_0$. This file should be names `betas_str.txt` and MUST be placed in the same directory as the out path specified for the run of the program. This file should contain a set of values in CSV format of dimension $p \times g$. It is a good idea to start these values at the marginal effects from multiple or marginal linear regression.

6.4 Parameters

Below is a canonical parameter submission file. We will outline the meaning of each of the components. It is noted that the program is currently only written for the $g = 2$ model and cannot, at this stage, handle problems with more components.

```
../../lmlr  /path/to/example/directory/simulation/sim_x.csv \
            /path/to/example/directory/simulation/sim_y.csv \
            10 \
            60 \
            1e-4 \
            500000 \
            1e-16 \
            2.2 \
            2.1 \
            -10 \
```

```
10 \
0.6 \
0.4 \
/path/to/example/output/directory/simulation/ \
2>&1 | tee /path/to/example/output/directory/simulation/simulation.log
```

In order we have file path to X matrix (requires full path), file path to Y vector (requires full path), lasso constraint Golden section search parameter lower bound, lasso constraint parameter upper bound, convergence criterion for the log-likelihood difference, maximum iterations, perturbation parameter first variance parameter σ_1^2, second sigma parameter σ_2^2, first intercept parameter α_1, second intercept parameter α_2, first pi parameter π_1, second pi parameter π_2, path to write output (requires full path) and is also the path to the starting betas (the starting betas must be named "betas_str.txt"), and the path for directing standard output to a text log file (requires that you just changes the path and leave $2 > \&1|$ `tee`.

7 Output

Along with the `.log` file, which captures all that is printed to standard output by the program, the program reports estimates for $\hat{\boldsymbol{\sigma}}$, $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\pi}}$, and $\hat{\boldsymbol{\beta}}$. Each are written in CSV text format with names `sigma_estimates.txt, alpha_estimates.txt, beta_estimates.txt, pi_estimates.txt`.

8 Commonly encountered erros

- `BOOST::Scale parameter is nan, but must be > 0 !` - FIX - This indicates that the you need to have better initial guesses of the component means or increase values for the variance components. Changing the initial bounds for the golden section search algorithm may also help.

9 References

References

Buhlmann, P., van de Geer, S., 2011. Statistics for High-Dimensional Data. Springer, New York.

De Veaux, R. D., 1989. Mixtures of linear regressions. Computational Statistics and Data Analysis 8, 227=245.

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society Series B 39, 1–38.

DeSarbo, W. S., Cron, W. L., 1988. A maximum likelihood methodology for clusterwise linear regressions. Journal of Classification 5, 249–282.

George, E. I., 2000. The variable selection problem. Journal of the American Statistical Association 95, 1304–1308.

Greene, W. H., 2003. Econometric Analysis. Prentice Hall.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements Of Statistical Learning. Springer, New York.

Hui, F. K. C., Warton, D. I., Foster, S. D., 2015. Multi-species distribution modeling using penalized mixture of regressions. Annals of Applied Statistics In Press.

Hunter, D. R., Li, R., 2005. Variable selection using MM algorithms. Annals of Statistics 33.

Izenman, A. J., 2008. Modern Multivariate Statistical Techniques. Springer, New York.

Jones, P. N., McLachlan, G. J., 1992. Fitting finite mixture models in a regression context. Australian Journal of Statistics 34, 233–240.

Khalili, A., 2010. New estimation and feature selection methods in mixture-of-experts models. Canadian Journal of Statistics 38, 519–539.

Khalili, A., 2011. An overview of the new feature selection methods in finite mixture of regression models. Journal of the Iranian Statistical Society 10, 201–235.

Khalili, A., Chen, J., 2007. Variable selection in finite mixture of regression models. Journal of the American Statistical Association 102, 1025–1038.

Khalili, A., Lin, S., 2013. Regularization in Finite Mixture of Regression Models with Diverging Number of Parameters. Biometrics 69, 436–446.

Lange, K., 2013. Optimization. Springer, New York.

Lloyd-Jones, L. R., Nguyen, H. D., McLachlan, G. J., 2016. A STUDY OF MAJOR LEAGUE BASEBALL SALARIES VIA A LASSO-PENALIZED MIXTURE OF LINEAR REGRESSIONS. Submitted to AOAS 0, 0–0.

Nguyen, H. D., McLachlan, G. J., 2015. Maximum likelihood estimation of Gaussian mixture models without matrix operations. Advances in Data Analysis and Classification In Press.

Quandt, R. E., 1972. A new approach to estimating switching regressions. Journal of the American Statistical Association 67, 306–310.

Stadler, N., Buhlmann, P., van de Geer, S., 2010. l_1-penalization for mxture regression models. Test 19, 209–256.

Tibshirani, R., 1996. Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society Series B 58, 267–288.

Wu, T. T., Lange, K., 2008. Coordinate descent algorithms for LASSO penalized regression. Annals of Applied Statistics 2, 224–244.

Yuan, M., Lin, Y., 2006. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society Series B 68, 49–67.