Lasso.jl Documentation

Release 0.0.1

Simon Kornblith

December 16, 2016

Contents

Contents:

CHAPTER 1

Lasso paths

fit (*LassoPath*, *X*, *y*, *d=Normal()*, *l=canonicallink(d); ...*)

Fits a linear or generalized linear Lasso path given the design matrix *X* and response *y*:

$$\underset{\beta}{\operatorname{argmin}} -\frac{1}{N}\mathcal{L}(y|X,\beta) + \lambda\left[(1-\alpha)\frac{1}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1\right]$$

The optional argument *d* specifies the conditional distribution of response, while *l* specifies the link function. Lasso.jl inherits supported distributions and link functions from GLM.jl. The default is to fit an linear Lasso path, i.e., *d=Normal(), l=IdentityLink()*, or $\mathcal{L}(y|X,\beta) = -\frac{1}{2}\|y - X\beta\|_2^2 + C$

Keyword arguments:

name	description	default
wts	Weights for each observation	`ones(length(y))`
offset	Offset of each observation	`zeros(length(y))`
α	Elastic Net parameter in interval [0, 1]. Controls the tradeoff between L1 and L2 regularization. α = 1 fits a pure Lasso model, while α = 0 would fit a pure ridge regression model. **Note**: Do not set α = 0. There are methods for fitting pure ridge regression models that are substantially more efficient than the coordinate descent procedure used in Lasso.jl.	`1`
λ, nλ, λminratio	Control the values of λ along path at which models are fit. λ can be used to specify a specific set of λ values at which models should be fit. If λ is unspecified, Lasso.jl selects nλ logarithmically spaced λ values from λ_{max}, the smallest λ value yielding a null model, to $\lambda\text{minratio} * \lambda_{\text{max}}$. If the proportion of deviance explained exceeds 0.999 or the difference between the deviance explained by successive λ values falls below 10^{-5}, the path stops early.	`nλ = 100` If more observations than predictors, `λminratio = 1e-4`. Otherwise, `λminratio = 0.001`.
standardize	Whether to standardize predictors to unit standard deviation before fitting.	`true`
intercept	Whether to fit an (unpenalized) model intercept.	`true`
algorithm	Algorithm to use. The NaiveCoordinateDescent algorithm, which iteratively computes the dot product of the predictors with the residuals, as opposed to the CovarianceCoordinateDescent algorithm, which uses a precomputed Gram matrix. NaiveCoordinateDescent is typically faster when there are many predictors that will not enter the model or when fitting generalized linear models.	NaiveCoordinateDescent if more than 5x as many predictors as observations or model is a GLM. CovarianceCoordinateDescent otherwise.
randomize	Whether to randomize the order in which coefficients are updated by coordinate descent. This can drastically speed convergence if coefficients are highly correlated, but is only supported under Julia 0.4.	`true` (if julia >= 0.4)
maxncoef	The maximum number of coefficients allowed in the model. If exceeded, an error will be thrown.	`min(size(X, 2), 2*size(X, 1))`
dofit	Whether to fit the model upon construction. If *false*, the model can be fit later by calling *fit!(model)*.	`true`
cd_tol	The tolerance for coordinate descent iterations iterations in the in-	`1e-7`

`fit` returns a LassoPath object describing the fit coefficients and values of λ along the Lasso path. The following fields are intended for external use:

field	description
λ	Vector of λ values corresponding to each fit model along the path
coefs	SparseMatrixCSC of model coefficients. Columns correspond to fit models; rows correspond to predictors
b0	Vector of model intercepts for each fit model
pct_dev	Vector of proportion of deviance explained values for each fit model
nulldev	The deviance of the null model (including the intercept, if specified)
nullb0	The intercept of the null model, or 0 if no intercept was fit
niter	Total number of coordinate descent iterations required to fit all models

For details of the algorithm, see Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1), 1.

Fused Lasso and trend filtering

`fit`(*FusedLasso*, *y*, λ)

Fits the fused Lasso model:

$$\underset{\beta}{\operatorname{argmin}} \frac{1}{2} \sum_{k=1}^{N} (y_k - \beta_k)^2 + \lambda \sum_{k=2}^{N} |\beta_k - \beta_{k-1}|$$

The model coefficients can be obtained by calling `coef` on the returned model object.

For details of the algorithm, see Johnson, N. A. (2013). A dynamic programming algorithm for the fused lasso and L0-segmentation. Journal of Computational and Graphical Statistics, 22(2), 246–260. doi:10.1080/10618600.2012.681238

`fit`(*TrendFilter*, *y*, *order*, λ)

Fits the trend filter model:

$$\underset{\beta}{\operatorname{argmin}} \frac{1}{2} \sum_{k=1}^{N} (y_k - \beta_k)^2 + \lambda \| D^{(k+1)} \beta_k \|_1$$

Where $D^{(k+1)}$ is the discrete difference operator of order k+1. The model coefficients can be obtained by calling `coef` on the returned model object.

For details of the algorithm, see Ramdas, A., & Tibshirani, R. J. (2014). Fast and flexible ADMM algorithms for trend filtering. arXiv Preprint arXiv:1406.2082. Retrieved from http://arxiv.org/abs/1406.2082

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Index

F