# Libra:
# R package for Linearized Bregman Algorithms in High Dimensional Statistics

Yuan Yao

School of Mathematical Sciences
Peking University

November 14, 2015

with Stanley Osher (UCLA), Feng Ruan (PKU & Stanford), Jiechao Xiong (PKU), and Wotao Yin (UCLA), et al.

# Cran R package: Libra (version 1.4)

`http://cran.r-project.org/web/packages/Libra/`

https://cran.r-project.org/web/packages/Libra/

**Libra: Linearized Bregman Algorithms for Generalized Linear Models**

Efficient procedures for fitting the regularization path for linear, binomial, multinomial, Ising and Potts models with lasso, group lasso or column lasso(only for multinomial) penalty. The package uses Linearized Bregman Algorithm to solve the regularization path through iterations. Bregman Inverse Scale Space Differential Inclusion solver is also provided for linear model with lasso penalty.

| | |
|---|---|
| Version: | 1.4 |
| Depends: | R (≥ 3.0), nnls |
| Suggests: | lars, MASS, animation |
| Published: | 2015-11-18 |
| Author: | Feng Ruan, Jiechao Xiong and Yuan Yao |
| Maintainer: | Jiechao Xiong <xiongjiechao at pku.edu.cn> |
| License: | GPL-2 |
| URL: | http://arxiv.org/abs/1406.7728 |
| NeedsCompilation: | yes |
| SystemRequirements: | GNU Scientific Library (GSL) |
| CRAN checks: | Libra results |

Downloads:

| | |
|---|---|
| Reference manual: | Libra.pdf |
| Package source: | Libra_1.4.tar.gz |
| Windows binaries: | r-devel: Libra_1.4.zip, r-release: Libra_1.4.zip, r-oldrel: Libra_1.4.zip |
| OS X Snow Leopard binaries: | r-release: not available, r-oldrel: not available |
| OS X Mavericks binaries: | r-release: Libra_1.3-2.tgz |
| Old sources: | Libra archive |

# Libra (1.4) includes

- linear regression
- logistic regression (binomial, multinomial)
- graphical models (Ising, Potts)

Two kinds of penalty:

- $l_1$-norm penalty(Lasso penalty)
- $l_2 - l_1$ penalty(Group Lasso penalty)

# Linear Regression

Linear Regression:

$$y = X\beta + \epsilon$$

Logistic Regression:

$$\frac{P(y=1|X)}{P(y=-1|X)} = e^{X\beta}$$

$\beta$ is sparse or group sparse, which corresponding two types of penalty.

- "ungrouped": $\sum_i |\beta_i|$
- "grouped": $\sum_g \sqrt{\sum_{g_i=g} \beta_i^2}$

# Linear Regression

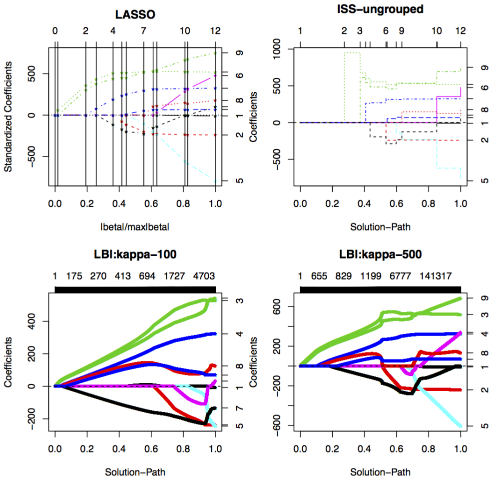
- Inverse Scale Space:
  $\texttt{iss}(\texttt{X}, \texttt{y}, \texttt{intercept} = \texttt{TRUE}, \texttt{normalize} = \texttt{TRUE},$
  $\texttt{nvar} = \texttt{min}(\texttt{dim}(\texttt{X})))$
- Linearized Bregman iteration:
  - kappa: damping factor
  - alpha: step size, satisfying $\texttt{alpha} \cdot \texttt{kappa} \|\Sigma_\texttt{n}\| = \texttt{c} \leq 2$

```
lb(X, y, kappa, alpha, c = 1, tlist, nt = 100, trate = 100,
  family = c("gaussian", "binomial", "multinomial"),
  group.type = c("ungrouped", "grouped", "columned"), index = NA,
  intercept = TRUE, normalize = TRUE)
```

# Example: Diabetes Data

```
data('diabetes')
attributes(x)
#$dim
# [1] 442  10
#$dimnames[[2]]
# [1] "age" "sex" "bmi" "map" "tc"  "ldl" "hdl" "tch"
    "ltg" "glu"

lassopath = lars(x,y)
isspath = iss(x,y)
lb(x,y,kappa=100,alpha=0.005,family="gaussian",group="
    ungrouped",intercept=FALSE,normalize=FALSE)
lb(x,y,kappa=500,alpha=0.001,family="gaussian",group="
    ungrouped",intercept=FALSE,normalize=FALSE)
```

# Example: Regularization Paths



# Another Example: ISS/LBI often beats LASSO

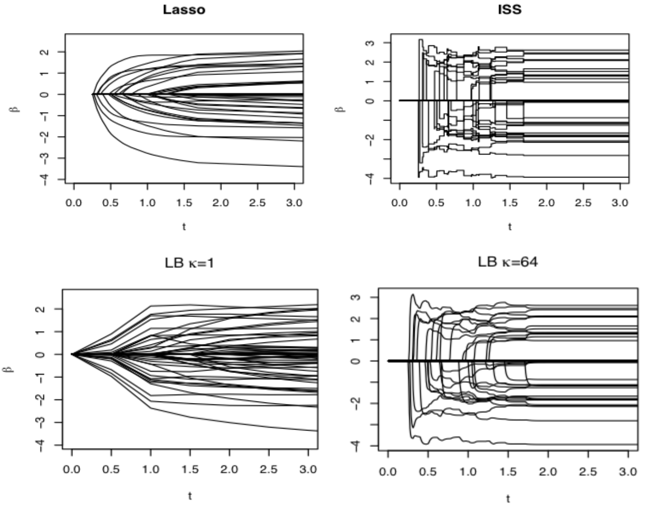
$n = 200$, $p = 100$, $S = \{1, \ldots, 30\}$, $x_i \sim N(0, \Sigma_p)$ ($\sigma_{ij} = 1/(3p)$ for $i \neq j$ and 1 otherwise)

| $\sigma$ | LB($\kappa = 4$) | LB($\kappa = 64$) | LB($\kappa = 1024$) | ISS | LASSO |
|---|---|---|---|---|---|
| 1 | 0.9771(0.0124) | 0.994(0.0069) | 0.9947(0.0065) | 0.9948(0.0064) | 0.9945(0.0068) |
| 3 | 0.9604(0.0169) | 0.9867(0.009) | 0.9882(0.0083) | 0.9884(0.0082) | 0.9879(0.0086) |
| 5 | 0.9393(0.0226) | 0.9659(0.0188) | 0.9673(0.0188) | 0.9676(0.0187) | 0.9671(0.0187) |

TABLE 1

*Mean AUC (standard deviation) for three methods at different noise levels ($\sigma$): ISS has a slightly better performance than LASSO in terms of AUC and as $\kappa$ increases, the performance of LB approaches that of ISS. As noise level $\sigma$ increases, the performance of all the methods drops.*

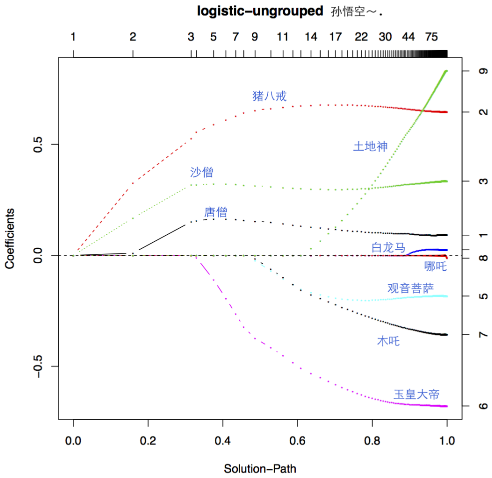
# But regularization paths are different



# Example: Journey to the West

```
load('~/data/XiYouJi10.RData')
attributes(data)
#$dim
#[1] 408   10
y<-2*data[,1]-1;
X<-as.matrix(2*data[,2:10]-1);

path <- lb(X,y,kappa=0.5,alpha=6,family="binomial",
           intercept=TRUE,normalize = FALSE,iter
              =300)
```

# Example: Regularization Paths



# Multinomial Logistic Regression

Multinomial Logistic Regression:

$$P(y=j|X) = \frac{e^{X\beta_j}}{\sum_i e^{X\beta_i}}$$

$\beta$ is k-by-p matrix.

- "ungrouped": $\sum_{i,j} |\beta_{ij}|$
- "columned": $\sum_i \sqrt{\sum_j \beta_{ij}^2}$
- "group": $\sum_g \sqrt{\sum_{g_i=g,j} \beta_{ij}^2}$

# Ising Model

$$P(x) \sim \exp\left(\sum_i \frac{a_{0i}}{2} x_i + \frac{1}{4}\sum_{i,j} \theta_{ij} x_i x_j\right)$$

where

- $\theta_{ij}$ the interaction coefficients
- $a_{0i}$ is the intercept coefficients
- Libra command:
  $\texttt{ising}(\texttt{X}, \texttt{kappa}, \texttt{alpha}, \texttt{c} = 4, \texttt{tlist}, \texttt{nt} = 100, \texttt{trate} = 100, \texttt{intercept} = \texttt{TRUE})$

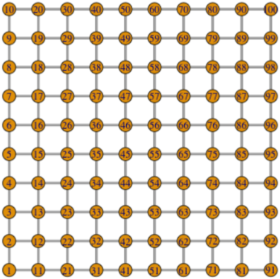
# Generalization: Potts Model

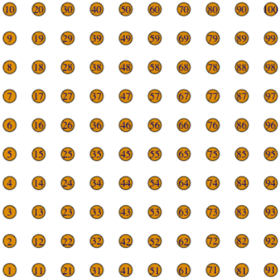
$$P(x) \sim \exp\left(\sum_{ip} a_{0,ip} 1(x_i = p) + \frac{1}{2}\sum_{ijpq} \theta_{ij;pq} 1(x_i = p) 1(x_j = q)\right)$$

where

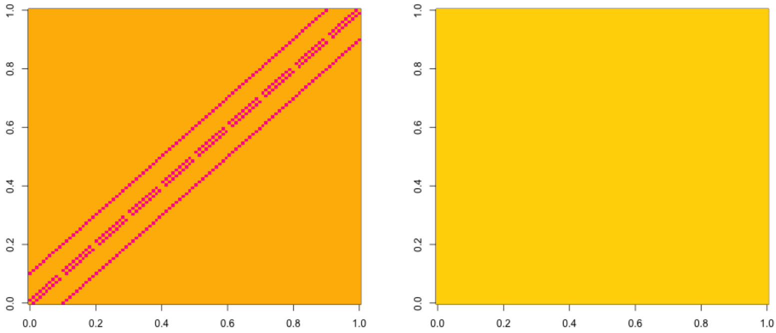
- $\theta_{ij;pq}$ the interaction coefficients
- $a_{0i}$ is the intercept coefficients
- Libra command:
  $\texttt{potts(X, kappa, alpha, c} = \texttt{1, tlist, nt} = \texttt{100, trate} = \texttt{100,}$
  $\texttt{type} = \texttt{c("entry", "block"), intercept} = \texttt{TRUE)}$

# Example: Ising Model Learning



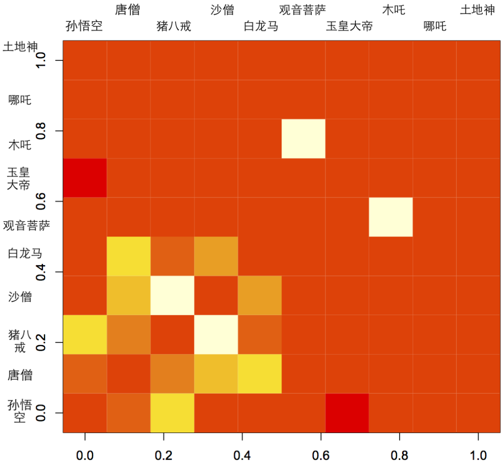
# Example: Ising Model Learning



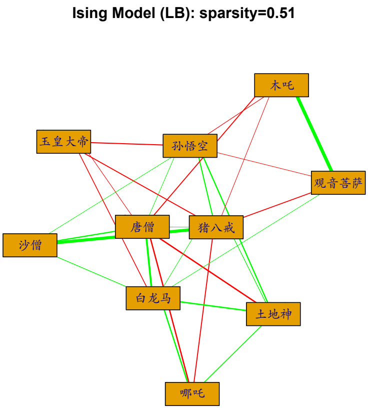
# Example: Ising Model of Journey to the West

```
load('XiYouJi10.RData')
attributes(data)
#$dim
#[1] 408   10
X<-as.matrix(2*data[,1:10]-1);

obj = ising(X,10,0.1,nt=1000,trate=100)
image(obj$path[,,500])


library('igraph')
g<-graph.adjacency(obj$path[,,850],mode="undirected",
    weighted=TRUE)
E(g)[E(g)$weight<0]$color<-"red"
E(g)[E(g)$weight>0]$color<-"green"
#plot(g,vertex.shape="rectangle",vertex.size=24,edge.
    width=2*abs(E(g)$weight))
V(g)$name<-attributes(data)$names
plot(g,vertex.shape="rectangle",vertex.size=25,vertex.
```

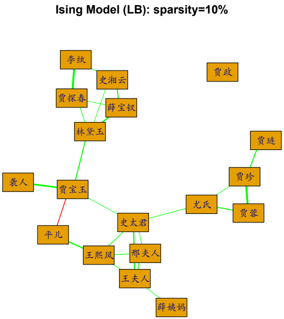
# Example: Ising Model of Journey to the West

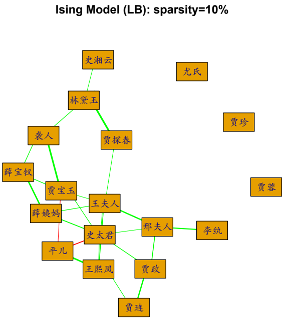


# Example: Ising Model of Journey to the West



# More Example: Dream of the Red Mansion (Xueqin Cao vs. E. Gao)



# Inside the R-package, (Linearized Bregman Iter.)

lies an *one-line* code essentially

$$z_{k+1} = z_k - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla_\theta L(x_i, \theta_k)$$

with

$$\theta_{k+1} = \kappa \cdot shrink(z_{k+1}, 1)$$

- $L(x, \theta)$ is the *loss* function to minimize
- $\alpha_k$ is step-size
- $\alpha_k \kappa \|\nabla_\theta^2 \hat{\mathbb{E}} L(x, \theta)\| < 2$
- $\theta_0 = z_0 = 0$

# Comparison with ISTA

- **ISTA**:

$$z_{t+1} = \textcolor{red}{Shrink}(z_t - \alpha_t X^T(Xz_t - y), \lambda)$$

- ISTA solves **LASSO** for fixed $\lambda$

$$\min_{\beta} \lambda_k \|\beta\|_1 + \frac{1}{2n}\|y - X\beta\|_2^2.$$

- parallel run ISTA for regularization paths of LASSO, $\lambda \in \{\lambda_k : k = 1, 2, \ldots\}$
- a single run of LB gives regularization path

# LB is the forward Euler discretization of

Damping dynamics

$$\dot{\rho}_t + \frac{1}{\kappa}\dot{\theta}_t = -\frac{1}{n}\sum_{i=1}^{n} \nabla_{\theta_t} L(x_i, \theta_t), \tag{1a}$$

$$\rho_t \in \partial \|\theta_t\|_1. \tag{1b}$$

starting at $t = 0$ and $\rho(0) = \theta(0) = 0$.

$$\rho_{k+1} + \frac{1}{\kappa}\theta_{k+1} = \rho_k + \frac{1}{\kappa}\theta_k - \frac{\alpha_k}{n}\sum_{i=1}^{n} \nabla_{\theta_k} L(x_i, \theta_k), \tag{2a}$$

$$\rho_t \in \partial \|\theta_k\|_1. \tag{2b}$$

or equivalently $z_{k+1} = z_k - \frac{\alpha_k}{n}\sum_{i=1}^{n} \nabla_{\theta_k} L(x_i, \theta_k)$

# $l_1$-Boost? Inverse Scale Spaces

Nonlinear ODE (differential inclusion) as $\kappa \to \infty$,

$$\dot{\rho}_t = -\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L(x_i, \theta_t), \quad (3a)$$

$$\rho_t \in \partial \|\theta_t\|_1. \quad (3b)$$

starting at $t = 0$ and $\rho(0) = \theta(0) = 0$.

- $L(x, \theta)$ is a loss function, e.g. negative log-likelihood
- piecewise-constant $\theta_t$
- $l_2$-Boost (Buhlman-Yu'02): $\dot{\theta} = -\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L(x_i, \theta_t)$

# Why dynamics?

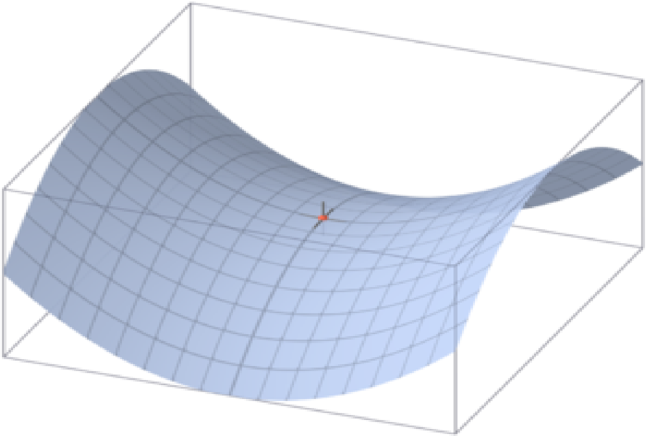
It exploits early stopping regularization to replace the $l_1$ regularization in LASSO

$$\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \nabla_{\theta} L(x_i, \theta) + \lambda \|\theta\|_1$$

yet boasting

- unbiased Oracle estimator at $t^* = 1/\lambda^*$ (Osher-Ruan-Xiong-Y.-Yin'14)
- every LASSO estimator is biased, nonconvex regularization (Fan-Li'01)

# Early stop around the saddle point



# Summary

The simple 1-line Linearized Bregman iteration:

- achieve mean path sign-consistency, **statistically equivalent to LASSO**
- and path sign-consistency with less bias, **better than LASSO**
- LB iteration is as simple as ISTA, but more powerful
  - cost: two free-parameters, $\kappa$ and step-size $\alpha_k$
  - tips: $\alpha_k \kappa \|\Sigma_n\| < 2$, large $\kappa$ to remove Elastic-net effect
- Early stopping regularization maybe better than penalization (e.g. Engl-Hanke-Neubauer'00, Y.-Rosasco-Caponnetto'07)
- A simple dynamics acts as if nonconvex optimization...

# For more: DSFA2015 tomorrow

- Talk 10:35-11:20, *L1Boost? A Dynamic Approach to Variable Selection and Sparse Recovery*
- Venue: A504/A510, Science Building at the North Zhongshan Road Campus, East China Normal University.

# Acknowledgement


- Theory
  - *Stanley Osher*, *Wotao Yin* (UCLA)
  - *Feng Ruan*, *Jiechao Xiong* (PKU)
- Applications: *Ming Yan* (UCLA); *Qianqian Xu*, *Chendi Huang* (PKU)
- Discussions: *Ming Yuan* (U Wisconsin), *Lie Wang* (MIT), *Peter Bickel* and *Bin Yu* (UCB)
- Grants:
  - IPAM, National Basic Research Program of China (973 Program), NSFC