Survival Data Analysis

Regularized Estimation in the Accelerated Failure Time Model

with High-Dimensional Covariates

H. Lee, S. Park

1. Introduction

Motivation of the paper

Cox PH Model		**AFT Model**
$\lambda(t; Z) = \lambda_0(t) \exp\{\beta' Z(t)\}$	→	$\log T_i = \beta' Z_i + \varepsilon_i$

About AFT Model

- It is used as alternative to the Cox model in the respect of intuitive linear regression interpretation
- It is used with an unspecified error distribution. That is Semi-parametric AFT
- Existing semi-parametric estimators (Buckley-James, rank-based estimator) have difficulties in **computing even when the number of covariates is relatively small**

1. Introduction

Purpose of the paper

For survival data with **high-dimensional covariates**,
finding covariates with good predictive power of survival is often
one of the most important aspect in the analysis

Unlike Cox model, no variable selection methods are available for the semiparametric AFT model

Main Topic of Paper

Two regularized version of Stute's weighted least squares(LS) estimator(Stute, 1993, 1996) in the AFT model with multiple covariates

1) **LASSO**(least absolute shrinkage and selection operator method)(Tibshirani, 1996)
2) **TGDR**(Threshold-Gradient-Directed Regularization method)(Friedman and Popescu, 2004)

2. Weighted Least Squares Estimation in AFT Model

AFT Model

$$T_i = \beta_0 + X_i'\beta + \varepsilon_i, \quad i = 1, \dots, n$$

T_i : logarithm of the failure time

X_i : a length-d covariate vector for the ith subject in a random sample of size n.

β_0 : the intercept

$\beta \in R^d$: the regression coefficient

ε_i : the error term.

Kaplan-Meier Weighted Least Squares

Following Stute and Wang(1993),
$\hat{F}_n$ can be written as $\hat{F}_n(y) = \sum_{i=1}^n w_{ni} 1 \left\{ Y_{(i)} \leq y \right\}$,
where the $w_{ni}'s$ are the jumps in the Kaplan-Meier estimator called
Kaplan-Meier weights

$$w_{n1} = \frac{\delta_{(1)}}{n}, \quad w_{ni} = \frac{\delta_{(i)}}{n-i+1} \prod_{j=1}^{i-1} \left(\frac{n-j}{n-j+1} \right)^{\delta(j)} \quad i = 2, \dots, n$$

Stute (1993,1996) proposed the weighted LS estimator $\hat{\theta} \equiv \left(\hat{\beta}_0, \hat{\beta} \right)$ that minimizes

$$L(\theta) = \frac{1}{2} \sum_{i=1}^n w_{ni} \left(Y_{(i)} - \beta_0 - X_{(i)}'\beta \right)^2$$

Use Kaplan-Meier weights to account for censoring in the LS criterion

3. Regularized Weighted LS Regression : LASSO

3.1 LASSO Estimator : WLS objective function

objective function

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{n} w_{ni}\left(Y_{(i)} - \beta_0 - X'_{(i)}\beta\right)^2$$

objective function of LASSO

$$L_\lambda(\theta) = \frac{1}{2}\sum_{i=1}^{n} w_{ni}\left(Y_{(i)} - \beta_0 - X'_{(i)}\beta\right)^2 + \boxed{\frac{\lambda}{n}\sum_{i=1}^{d}|\beta_j|}$$

where λ is a penalty parameter.

Weighte
d
mean
centering

Intercept = 0

objective function

$$L(\beta) = \frac{1}{2}\sum_{i=1}^{n}\left(Y_{w(i)} - X'_{(i)}\beta\right)^2$$

objective function of LASSO

$$L_\lambda(\beta) = L(\beta) + \boxed{\frac{\lambda}{n}\sum_{j=1}^{d}|\beta_j|}$$

$$L_\lambda(\beta) = L(\beta) \quad \boxed{s.t \quad \sum_{j=1}^{d}|\beta_j| \le u}$$

for a tuning parameter u

3.1 LASSO Estimator

Literature Reviews : L_1 boosting

Mason et al. (2000) and Friedman (2001) showed that boosting can be understood as a **gradient descent method** in a function space.
(i.e final model of boosting is a **linear combination** of base learners)

Mason et al. (2000) developed a gradient descent boosting to find the **optimal convex combination of base learners** by applying **Boosting on the convex hull** and called it "**L_1 boosting**".

Final model is given as $\sum_{n=1}^{\infty} w_n f_n$
with **L_1 regularization : $w_n \geq 0$ and $\sum_{n=1}^{\infty} w_n = 1$**

However, this algorithm does not work well for regression problems.

Kim et al. (2002) and Kim (2003) proposed two heuristic regularized boosting algorithms on the convex hull of base learners.

Kim, Y and Kim, J(2004) proposed **gradient descent algorithm for LASSO based on the L_1 boosting**

3.1 LASSO Estimator

Gradient descent algorithm for LASSO based on L_1 boosting

Problem in LASSO

the objective function is not differentiable

Special optimization technique

Tibshirani(1996)

QR (Quadratic Program) for LS

iteratively reweighted LS procedure with QP

For large dataset with high dimension

Kim, Y and Kim, J(2004)

Gradient descent algorithm for LASSO

Properties

1) It is **computationally simpler and faster** than the standard QP algorithm even though it is less accurate than QP or nonlinear algorithm
2) Converge rate is **independent on the dimension of input**
(less iteration gives more spare solution)
=> It is well suited with problems with large dimension inputs
3) But the convergence speed is rather slow at the near optimum

3.1 LASSO Estimator

L_1 boosting in optimizing LS of LASSO

Let $\boldsymbol{w} = \frac{\beta}{\lambda}$ and $\mathcal{S} = \boxed{\{\boldsymbol{w} : \|\boldsymbol{w}\|_1 \le 1\}}$ L_1 norm regularization

Optimization problem of generalized LASSO Convex set

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w} \in \mathcal{S}} C(\boldsymbol{w})$$

$$where\ C(\boldsymbol{w}) = R(\lambda \boldsymbol{w}) = R(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathcal{L}(y_i, \boldsymbol{x}_i, \boldsymbol{\beta}) = \sum_{i=1}^{n} (y - \boldsymbol{x}'\boldsymbol{\beta})^2 \cdots (1)$$

Main idea of the CGD algorithm to find $\hat{\boldsymbol{w}}$ sequentially

$$\boxed{\boldsymbol{w}[\alpha, \boldsymbol{v}] = \boldsymbol{w} + \alpha(\boldsymbol{v} - \boldsymbol{w})} \cdots (2)$$

① Search a direction vector $\boldsymbol{v}$ in $\mathcal{S}$ such that $C(\boldsymbol{w}[\alpha, \boldsymbol{v}])$ decreases most rapidly

$$C(\boldsymbol{w}[\alpha, v]) \approx C(\boldsymbol{w}) + \alpha \langle \nabla(\boldsymbol{w}), v - \boldsymbol{w} \rangle \cdots (3)$$

$$where\ \nabla(\boldsymbol{w}) = (\nabla(\boldsymbol{w})_1, \ldots, \nabla(\boldsymbol{w})_p)\ and\ \boxed{\nabla(\boldsymbol{w})_k = \partial C(\boldsymbol{w}) / \partial w_k.}$$

$$\min_{v \in \mathcal{S}} \langle \nabla(\boldsymbol{w}), \boldsymbol{v} \rangle = \min_{k=1,\ldots,p} \min\{\nabla(\boldsymbol{w})_k, -\nabla(\boldsymbol{w})_k\} \cdots (4)$$

the desired direction is a vector in R^p such that $\hat{k}$ th element is $\boxed{-sign(\nabla(\boldsymbol{w})_{\hat{k}})}$ and the other elements are zeros, where

$$\boxed{\hat{k} = \arg\min_{k=1,\ldots,p} \min\{\nabla(\boldsymbol{w})_k, -\nabla(\boldsymbol{w})_k\}} \cdots (5)$$

② Updates $\boldsymbol{w}$ to $\boldsymbol{w}[\alpha, \boldsymbol{v}]$.

1. Initialize: $\boldsymbol{w} = 0$ and $m = 0$.
2. Do until converge
   (a) $m = m + 1$.

① Search a direction vector

   (b) Compute the gradient $\nabla(\boldsymbol{w})$.

   (c) Find the $(\hat{k}, \hat{\gamma})$ that minimizes $\gamma \nabla(\boldsymbol{w})_k$ for $k = 1, \ldots, p$ and $\boxed{\gamma = \pm 1}$

   (d) Let $\boldsymbol{v}$ be the p dimensional vector such that the $\hat{k}$ -th element is $\hat{\gamma}$ and the other column elements are zeros.

② Search an update step size

   (e) Find $\hat{\alpha} = \arg\min_{\alpha \in [0,1]} C(\boldsymbol{w}[\alpha, \boldsymbol{v}])$

③ Update estimation

   (f) Update $\boldsymbol{w}$:

$$w_k = \begin{cases} (1 - \hat{\alpha}) w_k + \hat{\gamma}\hat{\alpha} & , k = \hat{k} \\ (1 - \hat{\alpha}) w_k & , k \neq \hat{k} \end{cases}$$

3. Return $\boldsymbol{w}$.

3.1 LASSO Estimator

L_1 boosting in optimizing Weighted LS of LASSO

1. Initialize: $\boldsymbol{w} = 0$ and $m = 0$.
2. Do until converge
   (a) $m = m + 1$.

   (b) Compute the gradient $\nabla(\boldsymbol{w})$.

   (c) Find the $(\hat{k}, \hat{\gamma})$ that minimizes $\gamma \nabla(\boldsymbol{w})_k$ for $k = 1, \ldots, p$ and $\gamma = \pm 1$

   (d) Let $\boldsymbol{v}$ be the p dimensional vector such that the $\hat{k}$ -th element is $\hat{\gamma}$ and the other column elements are zeros.

   (e) Find $\hat{\alpha} = \arg\min_{\alpha \in [0,1]} C(\boldsymbol{w}[\alpha, \boldsymbol{v}])$

   (f) Update $\boldsymbol{w}$:

$$w_k = \begin{cases} (1-\hat{\alpha})w_k + \hat{\gamma}\hat{\alpha} & , k = \hat{k} \\ (1-\hat{\alpha})w_k & , k \neq \hat{k} \end{cases}$$

3. Return $\boldsymbol{w}$.

---

1. Initialize $\beta^{(0)} = (0, \ldots, 0)'$. Let $m = 1$.
2. Do until convergence or a fixed number of iterations M has been reached

   (a) With the current estimate of $\beta^{(m-1)}$, compute $\boldsymbol{g}\left(\beta^{(m-1)}\right)$ the negative derivative of $L(\beta)$ with respect to β evaluated at $\beta^{(m-1)}$. Denote the jth component of $\boldsymbol{g}\left(\beta^{(m-1)}\right)$ as $g_j\left(\beta^{(m-1)}\right)$

   (b) Find that minimizes $\min\left\{g_j\left(\beta^{(m-1)}\right), -g_j\left(\beta^{(m-1)}\right)\right\}$.
   If $g_{j^*}\left(\beta^{(m-1)}\right) = 0$, then stop the iteration.

   (c) Otherwise, denote $\gamma = -sign\left(g_{j^*}\left(\beta^{(m-1)}\right)\right)$.
   Find $\hat{\kappa} \in [0,1]$ that minimizes $L\left((1-\kappa)\beta^{(m-1)} + \kappa \times u \times \gamma\eta_{j^*}\right)$, where η_{j^*} is a length-d vector that has the j^* th element equal to 1 and the remaining components equal to 0.

   (d) For the jth component of $\beta^{(m)}$,

LASSO penalty constraint

$$\beta_j^{(m)} = \begin{cases} (1-\hat{\kappa})\beta_{j^*}^{(m-1)} + \hat{\kappa}\gamma u & , j = j^* \\ (1-\hat{\kappa})\beta_j^{(m-1)} & , j \neq j^* \end{cases}$$

   (e) Replace m by $m + 1$

3. Return β

3.2 Threshold Gradient Directed Regularization

Gradient Descent method

1. Set $\nu = 0$
2. Start at a point in the parameter space $\hat{\beta}(\nu)$
3. "Descend" to the next point on the path via the update rule

$$\hat{\beta}(\nu + \Delta\nu) = \hat{\beta}(\nu) + \Delta\nu\, g(\nu)$$

where $\Delta\nu$ is an increment and $g(\nu)$ is the gradient of the empirical risk (i.e. average loss). In the case of linear regression, we have

$$g(\nu) = -\frac{d}{d\vec{\beta}}\frac{1}{N}\sum(y_i - X_i\beta)^2$$

evaluated at $\vec{\beta} = \hat{\beta}(\nu)$.

Consider
1,⋯, d
negative gradient

Friedman and Popescu(2004)

$$\mathrm{h}(\nu) = \{h_j(\nu)\}_0^d = \{\boxed{f_j(\nu)} \cdot g_j(\nu)\}_0^d$$

$$f_j(\nu) = I[|g_j(\nu)| \geq \tau \cdot \boxed{\max_{0\leq k\leq d} |g_k(\nu)|}]$$

where $0 \leq \tau \leq 1$

τ : threshold parameter.
It regulates the diversity of the values of $\{f_i\ (\nu)\}_0^d$

3.2 Threshold Gradient Directed Regularization

TGDR in AFT model

Δ : a fixed small positive number
m : iteration index, where $m \in \{0, 1, \ldots\}$
$\beta^{(m)}$: parameter estimate at the m th iteration.
τ : any fixed threshold value, where $\boxed{0 \leq \tau \leq 1}$

1. Initialize $\beta^{(0)} = (0, \ldots, 0)'$ and $\nu_0 = 0$. Let $m = 1$
2. With the current estimate of $\boxed{\beta^{(m-1)}}$, compute $\boxed{\mathrm{g}\left(\beta^{(m-1)}\right)}$, the negative derivative of $L(\beta) = \frac{1}{2}\sum_{i=1}^{n}\left(Y_{w(i)} - X'_{(i)}\beta\right)^2$ with respect to β evaluated at $\beta^{(m-1)}$. Denote the j th component of $\mathrm{g}\left(\beta^{(m-1)}\right)$ as $g_j\left(\beta^{(m-1)}\right)$. If $\max_j \left|g_j\left(\beta^{(m-1)}\right)\right| = 0$, stop the iteration.
3. Compute the vector $\mathrm{f}\left(\beta^{(m-1)}\right)$ of length d, where the j th component of $\mathbf{f}\left(\beta^{(m-1)}\right)$ is $\boxed{f_j\left(\beta^{(m-1)}\right) = I\left\{\left|g_j\left(\beta^{(m-1)}\right)\right| \geq \tau \max_l \left|g_l\left(\beta^{(m-1)}\right)\right|\right\}}$.
4. Update $\boxed{\beta^{(m)} = \beta^{(m-1)} + \Delta \mathrm{g}\left(\beta^{(m-1)}\right) \mathrm{f}\left(\beta^{(m-1)}\right)}$ and replace m by $m+1$.
5. Steps $2-4$ are repeated M times, where M is determined by cross-validation as described below.

3.3 Tuning Parameter Selection

LASSO	TGDR
(u)	(M, τ)
minimizing the AIC score $AIC\ score = \log(CV\ score) + \frac{2K}{n}$ where K = # of nonzero coefficient in $\hat{\beta}$	① M : for a fixed τ, minimize the CV score $CV\ score = \sum_{v=1}^{V} \left[L\left(\hat{\beta}^{(-v)}\right) - L^{(-v)}\left(\hat{\beta}^{(-v)}\right) \right],$ ② τ : minimize AIC score using the M determined in ① $AIC\ score = \log(CV\ score) + \frac{2K}{n}$

4. Asymptotic Properties of the LASSO Estimator

Stute(1993, 1996) + Knight and Fu(2000)

drive the asymptotic distribution of the Stute estimator under the L1 penalty

Assumptions:

A1. $E(\varepsilon \mid X) = 0$ and $E(T^2)$ is finite;

A2. T and C are independent and $P(T \leq C \mid T, X) = P(T \leq C \mid T)$

A3. $E(ZZ')$ is finite and nonsingular;

A4. $\tau_T < \tau_C$ or $\tau_T = \tau_C = \infty$

A5. (a) $E\left[(Y - Z'\theta^*)^2 ZZ'\delta\right] < \infty$

(b) $\int |(w - z'\theta^*) z_j| \times D^{1/2}(w)\tilde{F}^0(dz, dw) < \infty, for\ j = 0, \ldots, d$ and $D(y) = \int_0^{y-}[(1 - H(w))(1 - G(w))]^{-1}G(dw)$

THEOREM 1: Suppose that assumptions A1-A5 hold and that $n^{-1/2}\lambda_n \to \lambda_0 \geq 0$. Let $\Sigma_0 = E(ZZ')$. Then $n^{1/2}(\hat{\theta} - \theta^*) \to_D \arg\min(Q)$ as $n \to \infty$. Here

$$Q(\mathbf{b}) = -\mathbf{b}'\mathbf{W} + \mathbf{b}'\Sigma_0\mathbf{b} + \lambda_0 \sum_{j=1}^{d} \left[b_j sgn\left(\beta_j^*\right) 1\left\{\beta_j^* \neq 0\right\} + |b_j| 1\left\{\beta_j^* = 0\right\}\right]$$

where $\mathbf{W} \sim N(0, \Sigma)$ with $\Sigma = Var\left\{\delta\gamma_0(Y)(Y - Z'\theta^*)Z + (1 - \delta)\gamma_1(Y; \theta^*) - \gamma_2(Y; \theta^*)\right\}$

5. Numerical Studies

5.1 Simulation Study I : Finite-Sample Comparison

$$n = 200 \ and \ d = 30$$

$$T = \beta_0 + X'\beta + \epsilon \ where \ \beta_0 = 0.5 \ and \ \epsilon \sim N(0, 0.25)$$

		censoring rate		
		30%		70%
		criterion in comparison		
Generate $X = AX^*$ X^* : each component of length- d vector X^* is independently distributed as Unif $[-1, 1]$.	**1**	$\beta_0, \cdots, \beta_{10} = 1$ $\beta_{11}, \cdots, \beta_{30} = 0$	4	$\beta_0, \cdots, \beta_{10} = 1$ $\beta_{11}, \cdots, \beta_{30} = 0$
		many Smaller covariates		
A : $d \times d$ matrix that is upper-diagonal with all diagonal components of 1 and off-diagonal components such that the pairwise correlation between the ith and the jth components of X is $0.5^{\lvert i-j \rvert}$.	**2**	$\beta_0, \cdots, \beta_{15} = 0.4$ $\beta_{16}, \cdots, \beta_{30} = 0.2$	**5**	$\beta_0, \cdots, \beta_{15} = 0.4$ $\beta_{16}, \cdots, \beta_{30} = 0.2$
		three equally important groups		
$X_i = U_1 + \epsilon_i, U_1 \sim N(0,1), \quad i = 1, \ldots, 5$ $X_i = U_2 + \epsilon_i, U_2 \sim N(0,1), \quad i = 6, \ldots, 10$ $X_i = U_3 + \epsilon_i, U_3 \sim N(0,1), \quad i = 11, \ldots, 15$ $X_i \sim N(0,1), i = 16, \ldots, 30$ where ϵ_i are i.i.d. $N(0, 0.01), i = 1, \ldots, 15$.	**3**	$\beta_0, \cdots, \beta_{15} = 1$ $\beta_{16}, \cdots, \beta_{30} = 0$	**6**	$\beta_0, \cdots, \beta_{15} = 1$ $\beta_{16}, \cdots, \beta_{30} = 0$

5. Numerical Studies

5.1 Simulation Study I : Finite-Sample Comparison

Table 1

Simulation study comparing different estimation approaches

Example (count)	LS MSE	LS Count	LASSO MSE	LASSO Count	TGDR MSE	TGDR Count
1 (10)	0.351	29.9	0.654	10.0	0.153	16.2
2 (30)	0.352	30.0	1.407	7.4	0.144	29.9
3 (15)	3.246	30.0	4.241	9.2	0.227	22.0
4 (10)	1.363	29.9	2.875	8.3	0.578	21.5
5 (30)	1.450	30.0	3.174	5.9	0.531	29.6
6 (15)	15.07	30.0	15.42	7.5	1.507	26.0

MSE: mean squared error. Count: average number of nonzero coefficients based on 100 replications.

	LASSO
MSE	larger than LS and TGDR TGDR < LS < LASSO
average number of nonzeoro coefficient	Overall, underestimate The underestimation is serious with large number of small covariates. (Example 2,5)

5. Numerical Studies

5.1 Simulation Study I : Finite-Sample Comparison

Table 1

Simulation study comparing different estimation approaches

Example (count)	LS MSE	LS Count	LASSO MSE	LASSO Count	TGDR MSE	TGDR Count
1 (10)	0.351	29.9	0.654	10.0	0.153	16.2
2 (30)	0.352	30.0	1.407	7.4	0.144	29.9
3 (15)	3.246	30.0	4.241	9.2	0.227	22.0
4 (10)	1.363	29.9	2.875	8.3	0.578	21.5
5 (30)	1.450	30.0	3.174	5.9	0.531	29.6
6 (15)	15.07	30.0	15.42	7.5	1.507	26.0

MSE: mean squared error. Count: average number of nonzero coefficients based on 100 replications.

	TGDR
MSE	smaller than LS and TGDR TGDR < LS < LASSO Especially, outperforms with highly correlated covariates
average number of nonzeoro coefficient	Overall, overestimate relatively more accurate estimate in all example

5. Numerical Studies

Characteristics of LASSO and TGDR

LASSO	TGDR
gradient-directed iterative algorithm	
increase in the direction of single covariates	increase in the direction of multiple covariates
more diverse absolute coefficient values	more diverse absolute coefficient values than ridge but less than the LASSO
estimate only arbitrary one of coefficient even when two covariates are highly correlated	similar estimates for strongly correlated covariates
underestimate the number of nonzero coefficients	overestimate the number of nonzero coefficients

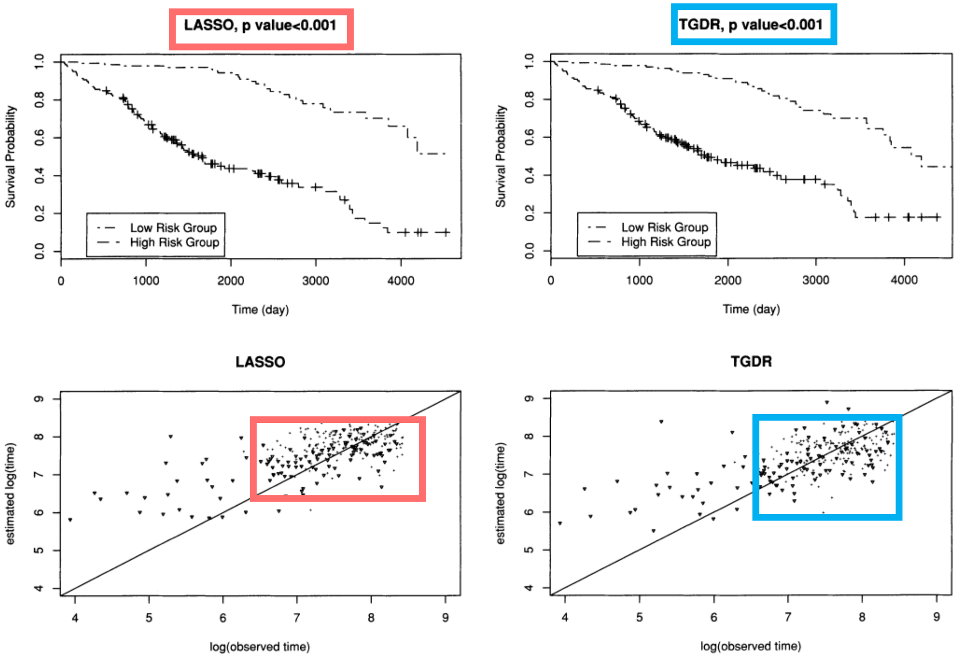
5. Numerical Studies

5.2 Simulation Study II : PBC Data

Table 3
PBC data

Covariate	LS Estimate	LS SE	LASSO Estimate	LASSO SE	TGDR Estimate	TGDR SE
Intercept	4.422	5.297	8.458	0.773	0.083	0.047
Age	−0.012	0.011	−0.009	0.010	−0.004	0.020
Alb	0.429	0.247	0.067	0.122	0.462	0.181
Log(alkphos)	0.166	0.132	0.003	0.022	0.292	0.295
Ascites	−0.490	0.637	0	0.427	−0.131	0.170
Log(bili)	−0.241	0.143	−0.268	0.105	−0.438	0.131
Log(chol)	0.246	0.281	0	0.007	0.290	0.210
Edtrt	−0.625	0.764	−0.982	0.727	−0.129	0.200
Hepmeg	0.195	0.189	0	0	0.020	0.107
Log(platelet)	−0.083	0.358	0	0.001	0.206	0.225
Log(protime)	1.084	1.417	0	0	0	0.013
Sex	0.045	0.339	0	0	0	0.094
Log(sgot)	−0.143	0.298	0	0	0.184	0.193
Spiders	−0.254	0.240	0	0.101	0	0.098
Stage	−0.123	0.093	−0.129	0.113	0	0.050
Trt	−0.039	0.161	0	0	0	0.080
Log(trig)	−0.132	0.250	0	0	0.178	0.208
Log(copper)	−0.148	0.131	0	0.024	−0.104	0.183

Estimate: estimated coefficient. SE: bootstrap standard error.



Conclusion

Properties of TGDR

Advantage	• It is capable of selecting a set of covariates that are highly correlated. • Simulation studies and the PBC data example showed the performance of TGDR on simultaneous variable selection and estimation.
Limitations	• It is a difficult problem to rigorously work out the approximate sampling distributions of the LASSO and the TGDR Stute estimators for cross-validated tuning parameters.
Extension	• Let U denote the vector of covariates of known importance and α its associated coefficients. $L(\alpha, \beta) = \frac{1}{2}\Sigma_{i=1}^{n} w_{ni}\left(Y_{(i)} - U'_{(i)}\alpha - X'_{(i)}\beta\right)^2$ • We can put the L1 penalty only on β to obtain a partial LASSO solution. We can also use TGDR on β only. • Furthermore, there may be models in which different penalties are appropriate for different parameters.

Reference

Kim, Y. and Kim, J.(2004). Gradient LASSO for feature selection. Proceedings of the 21st International Conference on Machine Learnings.

Kim, J. Kim,Y. and Kim,Y(2008). A Gradient-Based Optimization Algorithm for LASSO. Journal of Computational and Graphical Statistics , December 2008, Vol. 17, No. 4 (December 2008), pp. 994-1009

J. Gui and H. Li(2005). Threshold Gradient Descent Method for Censored Data Regression with Applications in Pharmacogenomics. Pacific Symposium on Biocomputing 10:272-283(2005)

Convex optimization problem : https://wikidocs.net/17206

Miller, J. M., Wallace, R. A., Smith, M. T., Lewis, R. V., Higgs, R. Q., Young, D. A., … Johnson, C. T. (2017). Trauma caretaking and compassion fatigue. *Trauma Prevention*, *14* (2), 243-45. doi: 10.XXXX.XXXXXX