3B1-R-2-7

乱択アルゴリズムを用いた特徴選択

A feature selection method based on randomized algorithm

杉本 和正[*1] 河原 吉伸[*1*2] 鷲尾 隆[*1*2]
Kazumasa Sugimoto Yoshinobu Kawahara Takashi Washio

[*1]大阪大学 産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

[*2]JST 科学技術振興機構
Japan Science and Technology Agency

In this paper, we propose a feature selection method based on randomized algorithm. As is well known, LASSO does not have the consistency for feature selection. To overcome this issue, we combined LASSO with pipage rounding technique, recently-proposed combinatorial algorithm for rounding. This approach is based on the fact that feature selection with least-squared errors is equivalent to submodular maximization. We give some empirical examples using artificial datasets, showing that our method could achieve better performance on feature selection.

1. はじめに

機械学習分野において特徴選択問題は頻繁に議論される重要な課題の一つである．訓練集合と呼ばれる既知のデータから予測モデルを生成し，そのモデルを用いて新たな入力データに対する出力を予測する問題は機械学習において主要なものであるが，このような問題を解く際に特徴選択を行なう事でデータの冗長性等を排除して予測計算の高精度化，高速化を図る事が出来る事が知られている．

近年機械学習分野では，特徴問題への有用なアプローチとして，LASSO などの正則化に基づく方法を中心に議論が行われている．しかし一般に，LASSO は特徴選択に関する一致性を持たない事が知られている [9]．つまり，正しく真の特徴を選択できる統計的性質は保証されない．この問題に対しては統計的立場からの議論が行われ AdaptiveLASSO などの方法が提案されているが，本稿では組合せ論的方法に基づきこの問題を解決するための方法について考察する．

組合せ論的手法として有名なものに貪欲法があるが，本研究ではそれとは異なる原理に基づいた組合せ論的な近似解法である乱択アルゴリズムを用いて特徴選択問題を考える．近年 Calinescu らによる連続貪欲算法とパイプ丸めという 2 つの乱択アルゴリズムを組み合わせたアルゴリズムが提案されている．しかし，この手法を計算機に実装すると次元数の増加につれて計算コストが指数関数的に増加し，実用性に欠ける事が判明した．この手法のうち，パイプ丸めの計算コストはそれほど高いものではなく，連続貪欲算法の計算コストが高かったため，今回はその連続貪欲算法を LASSO に置き換える事で計算コストを下げて特徴選択を行えるのではないか，というアイデアを元に研究を行った．この LASSO とパイプ丸めを組み合わせた手法を計算機に実装し，人工データを用いてその特徴選択精度を評価し，考察する．

本稿の構成は以下のようである．まず，2. で研究背景として特徴選択問題と LASSO について述べる．続いて 3. で Calinescu らが提案した連続貪欲算法とパイプ丸めを組み合わせた手法の概要と，今回提案する LASSO とパイプ丸めを組み合わせた手法の概要を述べる．4. にて人工データを用いた評価実験の結果と考察を述べ，5. でまとめとする．

連絡先: 杉本 和正，大阪大学 産業科学研究所，567-0047 大阪府茨木市美穂ヶ丘 8-1，sugimoto@ar.sanken.osaka-u.ac.jp

2. 研究背景

2.1 特徴選択問題

機械学習における予測手法とは，元となるデータから予測モデルを生成して新しい入力データに対する出力を予測するアプローチである．まず訓練集合と呼ばれる学習用のデータ $\{\boldsymbol{x}_i, y_i\}_{i=1}^n (n : \text{データ数})$ を用いて学習を行う．機械学習を行う際，学習の中で冗長なデータを削除したり，与えられたデータのうちの重要な部分だけを抽出する等してデータの多様性を減らす事が出来る．この過程を特徴選択 (特徴抽出) と呼ぶ．上の例で言うと $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id}) (d : \text{次元})$ の要素が特徴であり，特徴選択は実際に学習を行う際に重要な $\boldsymbol{x_i}$ の要素を抽出していく事となる．モデルの作成に用いるために集められたデータにはそのモデルにとって意味の無い情報が含まれている事が多く，そのような意味の無いデータはノイズとなってモデルの精度の低下，モデルサイズの肥大化，計算時間の増加等を招く．特徴選択を行うことでこのノイズを削除して有用な情報だけを用いて機械学習を行う事が出来る．特徴選択によるメリットとして，過学習の抑制，汎化誤差の減少による予測精度の向上，計算の高速化等が上げられる．

2.2 正則化学習による特徴選択手法

正則化学習の手法は様々だが，その中でも L1 罰則項を付与して罰則付き最適化を行うことでスパース解を得る手法である LASSO[7] に着目する．与えられた入力データ $\{\boldsymbol{x}_i, y_i\}_{i=1}^n (\in \mathbb{R}^d \times \mathbb{R})$ からパラメータ $\boldsymbol{\beta} (\in \mathbb{R}^d)$ を推測する問題を考える際，通常の二乗誤差最小化に対する正則化学習においては，誤差関数に罰則項を付与して最適化を行う事が知られている．

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \|y_i - \beta^T \boldsymbol{x}_i\|^2 + \frac{\lambda}{q} \sum_{j=1}^{d} |\beta_j|^q \qquad (1)$$

第二項が罰則項であるが，LASSO では L1 罰則項と呼ばれる項を付与する．

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \|y_i - \beta^T \boldsymbol{x}_i\|^2 + \lambda \sum_{j=1}^{d} |\beta_j| \qquad (2)$$

ここで λ はチューニングパラメータと呼ばれ，LASSO 解のスパース性を制御する重要な役割を果たす．λ の値を適切に定めて最適化を行うことでスパース解が得られる事になる．一般的

には式 (2) の形で用いられるが，厳密には LASSO の解は式 (3) で与えられる．

$$\min_{\beta\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \|y_i - \beta^T \boldsymbol{x}_i\|^2 \text{ s.t.} \sum_{j=1}^{d} |\beta_j| \leq t \tag{3}$$

ここで，$t(\geq 0)$ は係数の縮小量をコントロールするチューニングパラメータであり，先の λ と同じような意味を持つ．最小誤差解を$\hat{\boldsymbol{\beta}}$とし，$t_0 = \sum|\hat{\beta}_j|$ とすると，$t < t_0$ なる t は LASSO の解$\boldsymbol{\beta}$を 0 へと縮小し，幾つかの要素を 0 にする．この結果，LASSO 解$\boldsymbol{\beta}$はスパース解となる．式 (2)(3) は等価であり，$\sum_{j=1}^{d}|\beta_j| \leq t$ なる制約を与えて正則化を行う事は，L1 罰則項を付与して正則化を行うことと同義である事が知られている [7].

この LASSO は特徴選択手法としても用いる事ができ，LASSO 解中のゼロ要素は特徴として選択されなかった要素で，非ゼロ要素が特徴として選択されていると見なす事が出来る．しかし，LASSO は特徴選択における一致性を持たないという欠点がある事も同時に知られている [9]．この特徴選択に関する一致性とは正しく真のゼロ係数を特定できるという性質を表し [5, 6]，LASSO はこの性質を持たない．また，LASSO により特徴が選択されたモデルが得られるが，対象データを真に生成しているモデルと比較すると，LASSO で選択している個数が多すぎるという問題点もある．そのため，本研究では LASSO で得た解をパイプ丸めで丸める，つまり選択個数を減らす作業を組み合わせる事で，この問題点を改良出来るのではないか，と考える．

3. 提案手法

本研究では，LASSO とパイプ丸めを組み合わせた特徴選択手法を提案する．基本的な考えは 3.2 で述べる Calinescu らの手法内での緩和計算を LASSO 等の別の方法と置き換えるというものである．

3.1 劣モジュラ関数

本研究では組合せ最適化に用いる集合関数を単調な劣モジュラ関数であると仮定する．母集団 V とその部分集合 $A, B \subseteq V$ に対し，$f(A\cup B)+f(A\cap B) \leq f(A)+f(B)$ を満たす集合関数 $f : 2^V \to \mathbb{R}$ を劣モジュラ関数という．同様に $A \subseteq B \subseteq V$ に対し，$f(A) \leq f(B)$ を満たす関数を単調関数という．

関数の劣モジュラ性は連続領域での凸性に対応する離散領域における概念である事が知られている [4]．つまり，組み合わせ最適化において集合関数が劣モジュラ性を持つ場合には連続領域における問題と同様に最大化最小化を考える事が出来る．この事から，今回考える特徴選択問題は以下のように定式化出来る．

$$\max f(S) \quad \text{s.t. } |S| \leq k \tag{4}$$

このようにサイズ制約 k の下で劣モジュラ性を持つ集合関数 f の最大化あるいは最小化を考える事で，選択された特徴の集合 S を求める．

3.2 Calinescu らによる単調劣モジュラ関数の最大化手法

近年，乱択アルゴリズムを用いて連続解を整数解に丸める手法であるパイプ丸めを用いた単調劣モジュラ関数の最大化手法が Calinescu らにより提案されている [2]．この手法の概要を Algorithm1 に示す．

Algorithm 1 Calinescu らによる単調劣モジュラ関数最大化手法 [2]

1: 緩和手法の一種である連続貪欲算法 [8] を用いて，マトロイド制約上で $(1-1/e)$ 近似緩和解 z^* を求める．
2: パイプ丸め [1] により，$f(A^*) \geq f(z^*)$ なる $A^*(\in V)$ の範囲で緩和解 z^* を整数解に丸める．

ここで e は自然対数の底 (ネイピア数，$e \simeq 2.718$) である．この手法で得られる整数解は理論上 $(1-1/e)$ 以上の近似精度が保証される．つまり，$f(S)$ を最適解に対応する関数値，$f(S^*)$ をこの手法で得られた解に対応する関数値とすると

$$f(S^*) \geq f(S)(1-\frac{1}{e}) \tag{5}$$

を満たす解が得られる．しかし，この手法は次元数 d が大きい場合，step1 の緩和計算に指数時間単位の計算が必要となり，計算効率が下がる事が判明した．

3.3 パイプ丸め

前節で述べた Calinescu らによる手法，および今回提案する手法で用いるパイプ丸めについて以下に述べる．パイプ丸めは入力された連続値ベクトルを整数値に丸め込む事で整数解を出力するアルゴリズムである．入力された連続値ベクトル $\vec{y^*}$ からランダムに選んだ小数値成分 (y_i, y_j) のうち，サイズ制約の下でどちらかが整数値になるまで一方の値を小さくし，他方の値を小さくする事を繰り返す事で全ての要素が整数値になるように丸め込む．

ここで，今回考える特徴選択問題は式 (4) のように定式化される．このサイズ制約条件は均一マトロイドの制約条件と等しい．均一マトロイドとは，有限集合 X と X の部分集合の集合 $\mathcal{I}$ から定義されるマトロイド $\mathcal{M} = (X, \mathcal{I})$ のうち，$\mathcal{I} = \{S||S| \leq k\}(k$: 正の整数) という形で表されるものを指す．

また，集合のランク関数とタイト集合を定義 3.3.1 のように定義する．定義 3.3.1 から補助定理 3.3.1 が導ける．

定義 3.3.1 (ランク関数，タイト集合)**.** *1.* 集合 A のランクを $rank_{\mathcal{M}}(A) = \max\{|S| : S \subseteq A, A \in \mathcal{I}\}$ と定義する．

2. $\vec{y} \in [0,1]^d$ の i 番目の要素を y_i とし，$y(A) = \sum_{i\in A} y_i$ とする．$y(A) = rank_{\mathcal{M}}(A)$ の時，集合 A は $\vec{y}$ に対してタイト集合であると定義する．

補助定理 3.3.1. 集合 A, B が共にタイト集合の時，$A \cup B$ と $A \cap B$ もタイト集合である．

パイプ丸めの基本的な考えは以下のようなものである．まず，入力の連続値ベクトル $y \in [0,1]^d$ の小数値要素から (y_i, y_j) をランダムに選び，少なくとも (y_i, y_j) のどちらかは整数値になるように $(y_i+\delta, y_j-\delta)$ か $(y_i-\delta, y_j+\delta)$ に置換する．これを繰り返す事で y の要素が全て整数値に丸め込まれる．Algorithm2,3 に全体のアルゴリズムを示す．

3.4 提案手法

先に述べた通り，本研究では 3.2 で述べた Calinescu らによる手法の緩和計算を LASSO に置き換えたもの，つまり LASSO で得た連続解をパイプ丸めで整数値に丸め込む手法を提案する．Algorithm4 に概要を示す．

まず全ての (i, j) について $x_{ij}\beta_j$ を計算し，$\beta_j < 0$ である j については x_{ij} の正負を入れ替える．このように調整し

Algorithm 2 Pipage Rounding

入力:小数値要素を持つベクトル $\vec{y} \in [0,1]^V$.

While $\vec{y}$ が要素に小数値を持つ **do**
  $T \ni$($\vec{y}$ の小数値成分に対応する V の要素)
  つまり．$\vec{y}$ の i 番目の要素が小数値であれば $i \in T$.
  While $T \neq \emptyset$ **do**
    T から i, j をランダムに選ぶ.
    (y^+, A^+) を **HitConstraint**$(\vec{y}, i, j)$ から求める.
    (y^-, A^-) を **HitConstraint**$(\vec{y}, j, i)$ から求める.
    If $y^+ = y^- = \vec{y}$
      $T \leftarrow T \cap A^+$
    Else
      $p \leftarrow \|y^+ - y\| / \|y^+ - y^-\|$
      確率 p で $\vec{y} \leftarrow y^-, T \leftarrow T \cap A^-$
      確率 $(1-p)$ で $\vec{y} \leftarrow y^+, T \leftarrow T \cap A^+$
  EndWhile
EndWhile
成分が全て整数値となった $y \in \{0,1\}^V \leftarrow \vec{y}$ を出力

Algorithm 3 HitConstraint

$\mathcal{A} = \{A \subseteq V : i \in A, j \notin A\}$ とする.
$\Delta = \min_{B \in \mathcal{A}}(rank_{\mathcal{M}}(B) - y(B))$ なる Δ と，それに対応する集合 $B \in \mathcal{A}$ を求める.
If $y_j < \Delta$
  $y_i \leftarrow y_i + y_j$, $y_j \leftarrow 0$, $A' \leftarrow \{j\}$
Else
  $y_i \leftarrow y_i + \Delta$, $y_j \leftarrow y_j - \Delta$, $A' \leftarrow B$
(y, A') を出力.

た x_{ij} を $\bar{x}_{ij}$ とする．同時に，β_j が非零である場合には，重み $w_j (:= |\beta_j|^{-1})$ と掛け合わせる．これにより，x_{ij} に対して LASSO で得られた元の解である β_j と，$\bar{x}_{ij}$ に対する調整された解 $\bar{\beta}_j$ が得られる．次に，$\bar{\beta}_j$ がタイト集合の条件を満たすように $k'/\sum_{i=1}^{d} \bar{\beta}_j$ を掛け合わせる．このようにして得られた緩和解を入力としてパイプ丸めを適用し，最終的な整数解を得る.

ここで，Calinescu らによる手法ではパイプ丸めで求めた整数解に対する関数値が連続貪欲算法で求めた連続解に対するそれより悪化しないことは証明されている [2]．しかし，本研究では連続貪欲算法を LASSO に置き換えているため，LASSO で得た解をパイプ丸めで丸めても関数値が悪化しないように調整する必要があり，上で述べた作業でその調整を行なっている．LASSO の最適解は式 (2)(3) で与えられる，もし $\beta_j \in [0,1]$ である場合，$\sum_{j=1}^{d} |\beta_j| \leq t$ なる制約は均一マトロイドの制約条件と見なす事が出来る．そのため，連続貪欲算法と同じくマトロイド制約上での連続解と見なす事ができ，パイプ丸めの入力として用いる事が出来る.

Algorithm 4 LASSO+Pipage Rounding

入力:$\{\boldsymbol{x}_i, y_i\}_{i=1}^{n} (\in \mathbb{R}^d \times \mathbb{R})$ サイズ制約 k'

LASSO により緩和解 $\beta \in [0,1]^d$ を得る.
$x_{ij}\beta_j (i = 1, \ldots, n)$ を要素とする新しい入力データ $\bar{\boldsymbol{x}}_i (i = 1, \ldots, n)$ を求める.
重み $w_j = k'/(|\beta_j| \sum_j \beta_j)$ を求め，パラメータ $\bar{\beta} = w_j \beta_j$ を定める.
$\bar{\beta}_j$ を入力としてパイプ丸めを行い，スパースな整数解を得る.

以上が本研究で用いる手法だが，この手法が特徴選択手法としてどの程度の精度を持っているかはわからない．そのため人工データを用いてその特徴選択性能を評価する.

4. 実験・考察

今回提案する LASSO とパイプ丸めを組み合わせた手法を計算機で実装し，人工データを元に特徴選択性能の評価実験を行ってその結果を考察する.

4.1 実験環境

実験に用いた計算機は Intel Xenon CPU E5607 2.27GHz のプロセッサ，96GB のメインメモリ，Windows7 Professional 64bit の OS が搭載された Dell 社製の Precision R5500 である．また，計算ソフトは MathWorks 社の MATLAB R2011b を用いた.

4.2 人工データ

実験で用いる人工データとしてスパースな訓練データ $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ と，データを生成している真の特徴集合を表す $\boldsymbol{J}$ を生成する．訓練データのデータ数は 500 個，次元は 100 次元とした．真の特徴集合 $\boldsymbol{J}$ はこの手法の特徴選択性能を測るために用いる.

4.3 実験内容

実験で用いる集合関数 f を以下のように定義する.

$$g(S) = \sum_{i=1}^{N} \{y_i - \boldsymbol{w}_S \boldsymbol{x}_S\}^2 \tag{6}$$

$$f(S) = g(\emptyset) - g(S) \tag{7}$$

$\boldsymbol{x}_S$ と $\boldsymbol{w}_S$ は，人工データ $\boldsymbol{x}$ と LASSO 解から求めた $\boldsymbol{w}$ から S に対応する要素のみを抽出したベクトル，行列である．この f は $f(\emptyset) = 0$ かつ単調な劣モジュラ関数である [3]．実験では，まず人工データに LASSO を適用して連続解を求め，その特徴選択精度を求める．LASSO で得られた解を 3. で述べたように調整してパイプ丸めにより整数解を求め，その特徴選択精度を LASSO の精度と比較を行う．比較方法は，得られた集合を S とすると「(S と J の共通要素の個数)/(J の個数)[%]」として求めた割合を比較する．これをサイズ制約 k' の値をいくつか変えて行う．なお，この手法は乱択アルゴリズムに基づくために実験結果にはばらつきが生じる．そのためそれぞれ複数回実験を行なっている.

4.4 結果・考察

図 1〜4 に実験結果を図示し，図 5 に求めた精度の一覧を示す．今回はサイズ制約 k の値を $k = 10, 20, 30, 40$ として実験を行い，各 k ごとに図を作成している．グラフ中では横軸に LASSO のスパース性 (100 個の要素中何個がゼロ要素か)[%] を取り，縦軸に精度 [%] を取っている．LASSO の解を求める際に λ の値を一定にして計算したため，横軸の LASSO の非ゼロ要素数に偏りが生じている．そのため，図によって横軸の領域を変えている．図中の赤い ◯ が LASSO の精度で，青い △ が提案手法の精度である．また，図 5 中では左から LASSO のスパース性が低い順にソートされている.

図から分かる通り，LASSO 解がある程度スパースな解になっている場合は提案手法により特徴選択精度の向上を見込める事がわかった．今回は LASSO 解のスパース性の制御を行なっ

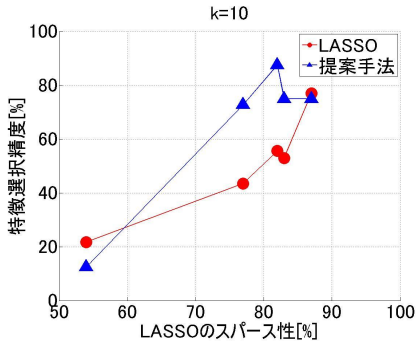

図 1: サイズ制約 10

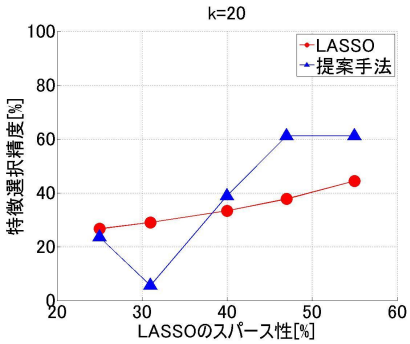

図 2: サイズ制約 20

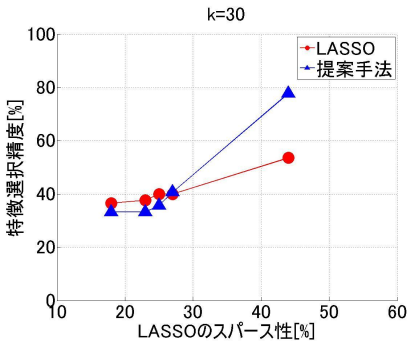

図 3: サイズ制約 30

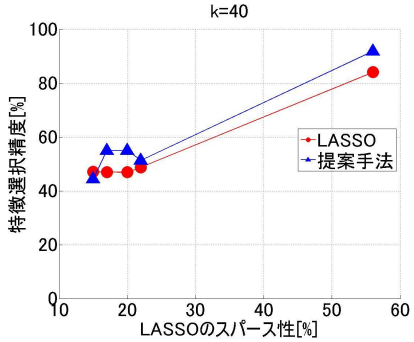

図 4: サイズ制約 40

k = 10	LASSO[%]	21.74	43.48	55.56	52.94	76.92
	提案手法 [%]	12.50	72.72	87.5	75.00	75.00
k = 20	LASSO[%]	26.67	28.99	33.33	37.73	44.44
	提案手法 [%]	23.53	5.56	38.89	61.11	61.11
k = 30	LASSO[%]	36.58	37.66	40.00	40.00	53.57
	提案手法 [%]	33.33	33.33	35.71	40.74	77.78
k = 40	LASSO[%]	47.06	46.99	46.91	48.72	84.09
	提案手法 [%]	44.44	55.00	55.00	51.35	91.89

図 5: 実験結果一覧

ていないため横軸の範囲が限定されてしまっているが，その範囲内でも精度を上げる事が出来ている．本来ならば横軸の値が 0〜100 まできれいに分布するように取り，例えば LASSO のスパース性が $0, 10, 20, 30, \ldots, 100$ 周辺になるような状態でそれぞれ複数回実験を行う等すれば更に提案手法の確実性を確認することが出来ると思われるが，LASSO のスパース性の制御が難しかったため今回は見送っている．

今後の課題としては，今回はこの手法の統計的性質については考察していないため，今回のような経験的な評価だけでなく，理論的な面からも精度の向上を保証する必要があると思われる．また，上で述べたように，サンプル数を増やす事でスパース性と精度の相関関係や，どのぐらいの確率で精度が上がるか等も確認していきたい．

5. まとめ

特徴選択問題において用いる集合関数が劣モジュラ関数である場合には，問題を最大化，最小化問題として考える事が出来る．劣モジュラ最大化問題として特徴選択問題を見た場合，既存の手法としては統計学的な観点から考案された手法が主であったが，本研究では組合せ論的な観点からその解法を考えた．今回は正則化学習に基づく手法である LASSO により得た連続解をパイプ丸めで整数解に丸めるという手法を提案した．人工データを用いた評価実験により，この手法では LASSO よりもある程度特徴選択精度の向上が見込まれる事がわかったが，LASSO よりも精度が下がってしまう場合もあり，まだ検討，修正の余地が残されている．

参考文献

[1] A. Ageev and M.I. Sviridenko. Pipage rounding: a new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, Vol. 8, No. 3, pp. 307–328, 2004.

[2] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput*, 2009.

[3] Abhimanyu Das and David Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pp. 45–54, New York, NY, USA, 2008. ACM.

[4] J. Edmonds. Submodular functions, matroids, and certain polyhedra. *Combinatorial Optimization?Eureka, You Shrink!*, pp. 11–26, 2003.

[5] J. Fan and R. Li. Variable selection for cox's proportional hazards model and frailty model. *The Annals of Statistics*, Vol. 30, No. 1, pp. 74–99, 2002.

[6] J. Shao. An asymptotic theory for linear model selection. *Statistica Sinica*, Vol. 7, pp. 221–242, 1997.

[7] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[8] L.A. Wolsey. Maximising real-valued submodular functions: Primal and dual heuristics for location problems. *Mathematics of Operations Research*, pp. 410–425, 1982.

[9] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, Vol. 101, No. 476, pp. 1418–1429, 2006.